Waseda University Doctoral Dissertation

Study on Hybridized Estimation of Distribution Algorithm with Probabilistic Graphical Models and Scheduling Applications

Xinchang HAO

Graduate School of Information, Production and Systems
Waseda University

September 2016

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Background

In a world of increasing global competitiveness, the current manufacturing must overcome challenges associated with high product mix, irregular demand patterns, customizable orders, and short product life cycles, particularly for electronic industry. Crucial factors of competitiveness drive industries to deploy Computer Numerical Controlled (CNC) machines assembled by Automated Material Handing Systems (AMHS) . These machines always need auxiliary resources to provide multiple features. The current manufacturing becomes more and more agile intending to thriving in an unpredictable environment and attracting global customers [1]. In this situation, improvement of operational processes is likely to major drivers to realize the necessary cost reduction and meet the reduced cycle time. Consequently, the scheduling dealing with product-mix processing, job transportation and configuration of machines is ultimately required for the Agile Manufacturing System (AMS) .

From optimization point of view, the traditional job-shop scheduling problem is a classical one of NP-hard discrete optimization problems. In job shop agile manufacturing environment, machines equipped with auxiliary resources (such as cutting tools for machining products) gives the flexibility of manufacturing system. herein, the problem is to determine how, when and in which sequence to effectively allocate these operations of jobs to the suitable manufacturing resources so as to achieve the given objectives while maintaining the feasibility of the process plan and schedule. This problem can be defined as follows:

- Task permutation (Operation sequencing): Determine the executing sequence of all the operations required for the jobs so that the precedence relationships among all the operations are not violated.

- Resource allocation: Determine how and when to assign the manufacturing machines to the jobs with regard to the job's constraints, as well as decide the accessory resources (such as tools) equipped in the machine.

In last decades, intelligent manufacturing planning and scheduling based on meta-heuristics, such as Simulated Annealing (SA) , Genetic Algorithms (GAs) and Binary Particle Swarm Optimization (BPSO) have grown into common tools for achieving sensible solutions within considerable computational times in real settings. Unfortunately, based on the traditional methods it is hard to deliver an effective solution in a limited computation time, especially for large size problems (referred as high-dimensional optimization problems).

Recently, Discrete Estimation of Distribution Algorithms (DEDA), a probabilistic method based stochastic optimization method, was proposed to solve the high-dimensional optimization problem and applies to many issues such as in computer science, electrical engineering, network engineering, industrial engineering, and management science [2] [3]. However, these approaches assume that the decision variable concerning with machine assignment is independent, and the optimality has significantly decreased for the problems with interdependence among the dimensional variables. It is essential to design effective computational methods to predict the structure and parameters between the variables concerning resource allocation.

On the other hand, machine learning has become one of the mainstays of information technology in artificial intelligence. Inherently, it has the capability to learn the relationship and structure between the variables based on the labeled data set (referred to promising data). The techniques has taken a great progression in many application areas including speech recognition, image or text classification, or bioinformatics problems [4].

1.2 Research objectives

In this dissertation, we focus two canonical types of flexible and responsible scheduling systems considering product-mix and machine/tool allocation with variety of small batch jobs.

- (**FJSP (Type 1)**) flexible job shop scheduling problem where the setup time of operation is negligible. It is one of the most popular scheduling models existing in practice which is among the hardest combinatorial optimization problems.

- (**FJSP (Type 2)**) flexible job shop scheduling problem where the setup time of operation is not negligible. There has been a significant increase in interest of scheduling problems involving setup times. Since there are tremendous savings when setup times/costs are explicitly incorporated in scheduling decisions in various real world industrial/service environments.

We propose advanced scheduling algorithm for solving two canonical types of scheduling problems in order to achieve:

-better optimality.

-better stability.

-better computation efficiency.

In this dissertation:

(1) develop software architecture of advanced EDA With Probabilistic Graphical Model.

(2) propose an Efficient Markov Network Field Learning Algorithm.

(3) propose an effective Bayesian Network Learning Algorithm.

(4) evaluate the proposal on benchmark of FJSP problems.

For scheduling problem in AMS, the interaction of the decision on resource allocation is crucial during the decision-making process. Therefore, a framework of Probabilistic Graphical Models (PGM) is proposed to predict the many-to-many interdependence among decision variables concerning resource allocation. Incorporating machine learning technique into meta-heuristics for combinatorial problems optimization techniques, we propose an novel hybridized EDA. Based on two different GPMs, two types of hybrid EDA are developed in this dissertation. One is Markov random field-based EDA (MREDA) for solving scheduling problems which includes combinational interdependence relationship among decision variables, and the other one is Bayesian network-based EDA (BSEDA) for solving Scheduling problems with sequence-dependent setup times where Bayesian network is employed to model the sequential interdependence relationship among decision variables concerning resource allocation. To evaluate merits of the proposed algorithm our approaches, we present the application of MREDA to solve well-studied scheduling problems: flexible job-shop scheduling (FJSP) and study the application of BSEDA to solve FJSP (Type 2) [5]. We proved the effectiveness of the proposed methods by comparative experiments with conventional meta-heuristic scheduling methods.

The major contribution is shown in Fig.1.1.

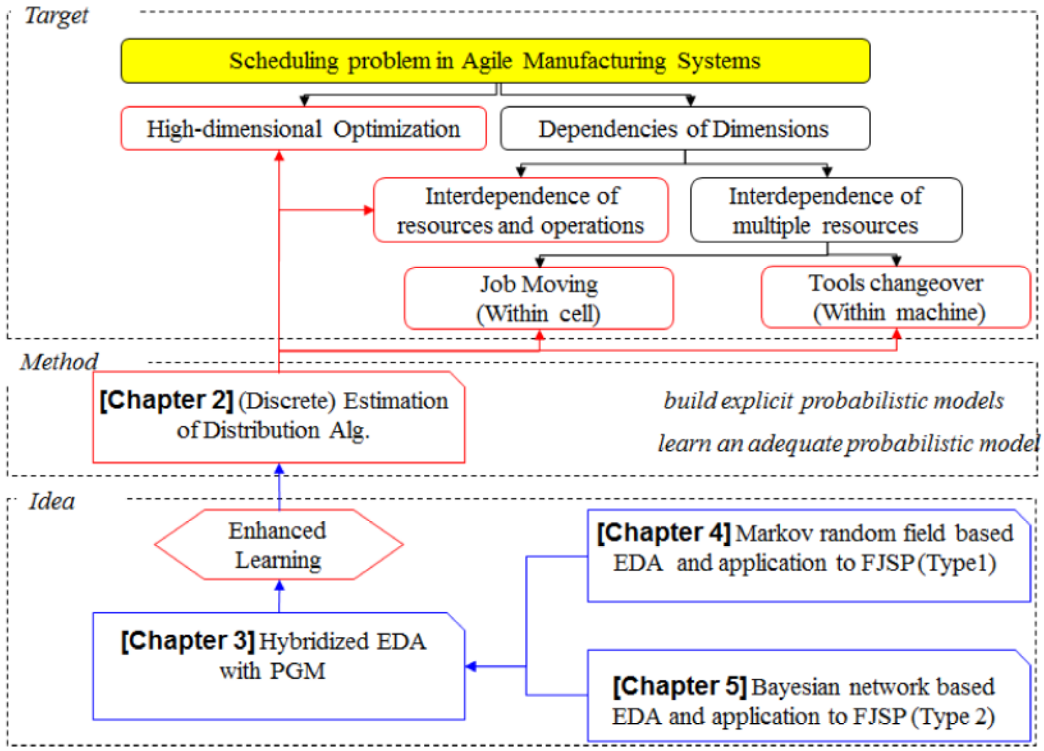


Fig. 1.1 Major contribution of this dissertation

1.3 Outline of this dissertation

Chapter 2 gives an overview of EDAs and the application related to scheduling problems. Particularly, we survey the previous approaches according to the probabilistic model. Next, we briefly introduce the literature of Probabilistic Graphical Model (Markov Random Field and Bayesian network) concerning machine learning techniques. Bayesian network can be used for modeling the cause and effect relationship between the decision variables. Markov network can be used to model interdependency of decision variables.

Chapter 3 For the scheduling problem of the manufacturing environment consists of multiple-functional machines providing flexibility for processing the jobs. Inherently, there are two types of decision interdependencies in resource allocation problems: (1) resource allocation to combination-dependency (2) sequence-dependency. However, this kind of many-to-many decision interdependencies in resource allocation are not efficiently considered in traditional EDAs. Therefore, we propose a novel EDA based on network- graphical model to construct those decision interdependencies in resource allocation problems.

Firstly, we proposed Markov random field-based EDA (MREDA). In MREDA, Markov network is used to model combination-dependency of decisions concerning resource allocation of operations. MREDA takes care of exploration that tries to identify the most promising search space regions, and to model conditional dependence of the resource allocation variables. The conditional probabilities defined by the local Markov property are estimated, and the new candidate solutions are sampled according to the given sampling method. For a candidate solution, problem specific based local search algorithm is used to improve each candidate solution to reach a local optimum.

Secondly, we propose Bayesian network based EDA (BSEDA). Different to the traditional chain, tree model, Bayesian network in BSEDA is employed to model the many-to-many sequence-dependencies of resource allocation. It can be more efficient to exploit resource allocation considering changoever along the job process routing. To enhance the ability to solve large-scale problems of high-dimensional optimization; Especially for local exploitation. Thus, we consider using two sub-populations based GA-based algorithm to adjust the machine assignment and operation sequence respectively with a splitting criterion and a combination criterion.

Chapter 4 presents the design of Markov random field-based EDA (MREDA) and its scheduling application. Flexible job-shop scheduling problem (FJSP) with minimizing make-span is conducted to evaluate the merit of the proposed MREDA. Wherein, FJSP is one of classical NP-hard problem expanded from the traditional job-shop scheduling problem (JSP). In solving FJSP, machine allocation to per operation in FJSP can be formulated as a many-to-many network model. Thus, Markov property described by an undirected graph can be efficiently applied to FJSP.

Chapter 5 studies the design of Bayesian network-based EDA (BSEDA) and its scheduling application. To evaluate the proposed BSEDA, Experiment is conducted to solve Flexible job-shop scheduling problem with setup-time considering minimizing make-span. Different with FJSP (Type 1), FJSP (Type 2) considers manufacturing environment with multi-purpose machines equipped with tool-box, and the tool changeover and part changeover have highly depended on job processing sequence. Bayesian network is employed to learn machine allocations of the operations affecting the make-span, simultaneously model the parenthood relationship between tools equipped with the machine.

Chapter 6 outlines the main results and contributions, and presents the possible extensions for the future researches.

Chapter 2

Related Work

2.1 EDAs and Application to Scheduling

There is an imitation of biological interest much stronger since 1960s, trying to solve the optimization problem of this type. Human being results of simulating the natural evolution process of stochastic optimization technique known as evolutionary algorithms (EAs), which can usually be difficult than conventional optimization method is applied to the reality of the problem. The EA mainly involves meta-heuristic optimizations such as genetic algorithm (GA) [6, 7] and other EAs [8–11]. Recently,some swarm-based EAs attract researchers' attention which contains particle swarm optimization (PSO) [12, 13] and ant colony optimization (ACO) [14] etc.

The behaviors of EAs largely depend on operators and the related parameters of crossover and mutation probability, population size and so on. Researchers need to experience the resolution and use of these algorithms, to select the appropriate values of these parameters. In addition, the best choice for selection task, the values of these parameters were suggested itself as an additional optimization problem [15]. In addition, the genetic algorithm according to some problems of under-performing existing operator, crossover and mutation is no guarantee that keep building block hypothesis.All these reasons to create a new type in the name of the classification algorithm of distribution algorithms (EDA) [16], trying to make easy to predict the movement of population in the search space and avoid needs of many parameters. These algorithms are based on the evolution process of search and the genetic algorithm, theoretical basis of probability theory. In brief, EDA population-based search algorithm is based on the probability model of promising solutions combined with the simulation model of induction to guide their search.

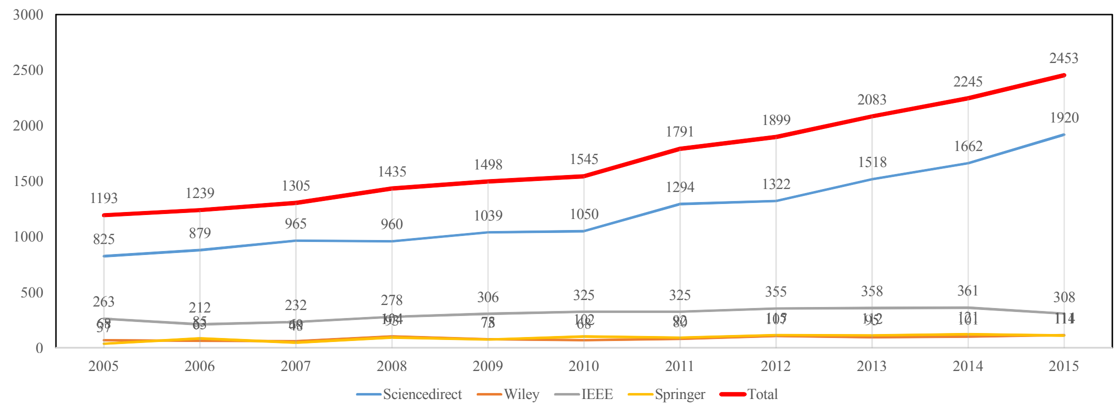


Fig. 2.1 Number of papers published on EDAs

2.1.1 Survey of EDAs

Discrete Estimation of Distribution Algorithms (DEDA), a probabilistic method based stochastic optimization method, was proposed to solve the high-dimensional optimization problem and applies to many issues such as in computer science, electrical engineering, network engineering, industrial engineering, and management science [2] [3]. As shown in Fig.2.1, the number of papers related EDAs grow at an average annual rate of 8% in recent 5 years. In 2015, there are 2500 papers published on the major publishers. In most of these applications, most researches stress that EDAs can achieve better performance, or can solving real-world large-scale size problems such as flow-shop scheduling problem. However, these approaches assume that the decision variable concerning with machine assignment is independent, and the optimality has significantly decreased for the problems with interdependence among the dimensional variables. It is essential to design effective computational methods to predict the structure and parameters between the variables concerning resource allocation.

Evolutionary Algorithms

As shown in follows, there are five basic parts of evolutionary algorithm summarized by Michalewicz [17]:

1. Individual with the expression of feasible solution of the problem.
2. Evaluation function with evaluate solutions by fitness values.
3. Evolutionary operations with alter the composition of individuals.
4. Population with construct an initial set of possible solutions
5. Algorithm parameters with adjust implementation process.

Let $P(t)$ be parents and $C(t)$ be offspring, let t be the current generation, respectively. Fig. 2.2 shows an evolutionary process of EA. The implementation process of EA is shown in Fig.2.3.

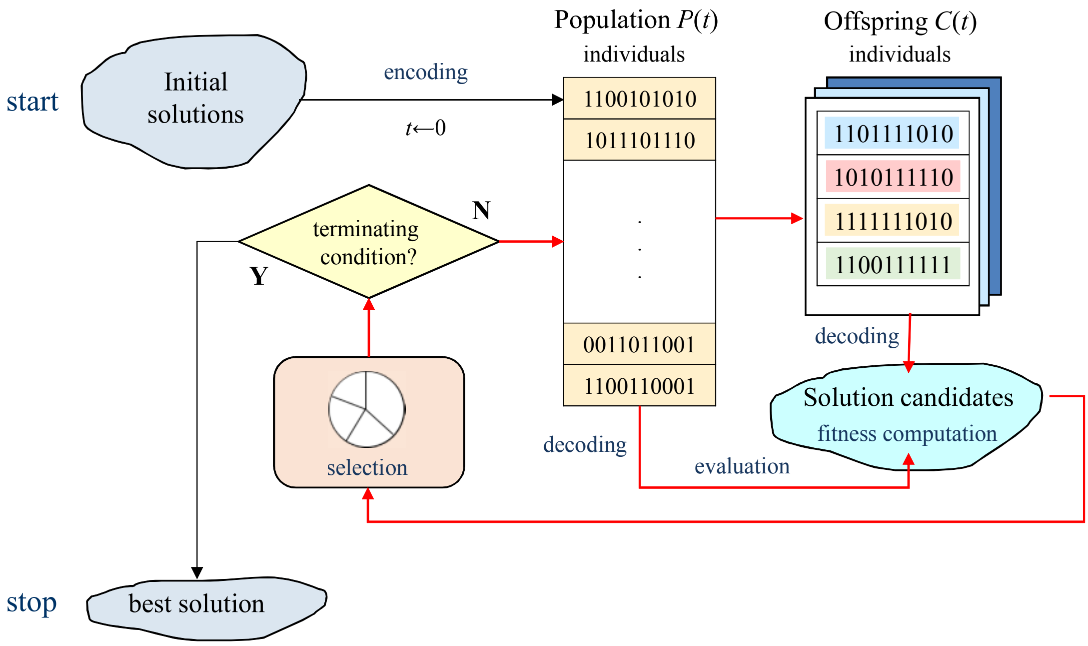


Fig. 2.2 The implementation process of evolutionary algorithm

Different with GA or other EAs, generating a new population of individuals do not use the crossover and mutation operator. Additionally, the new personal sampling from the probability distribution estimation from database contains only individual generation to choose from. At the same time, in other heuristic EA summit on behalf of the relationship among different variables of the individual is to remember the implicit (such as a building block hypothesis), in the EDA the interrelations is through the joint probability distribution associated with personal choice in each iteration. Estimating tasks related to the database, in fact, the choice of joint probability distribution of individual from a previous generation is the most difficult execution of work. In particular, the latter need to adapt to the learning model data, the researchers developed a probabilistic graphical model of the domain.

Early EAs, encoding was combined by using binary strings. Base on the binary encoding, the function optimization has serious defects, the existence of hamming cliffs. For example, belong to the phenotype space, 011111 and 100000 are very close, but in the genotype space, it has largest hamming distance. All through the hamming cliffs must be changed. Therefore the binary code cannot save the point position of phenotype space. For many optimization

Algorithm 1: general Evolutionary Algorithm

```
Input: problem data, parameters
Output: the best solution
begin
    t ← 0;
    initialize P(t);    // P(t): parents in generation t
    evaluate P(t);
    while not terminating condition do
        create C(t) from P(t);    // C(t): offspring in generation t
        evaluate C(t);
        select P(t+1) from P(t) and C(t);
        t ← t+1;
    end
    Output the best solution
end
```

Fig. 2.3 Pseudo-code of general Evolutionary Algorithm

problems, binary string-based code cannot effectively represent the solutions of problem. How to represent the solutions of the problem by various codes will affect the effectiveness of EAs. There are classified code types as follows:

- Binary string-based code
- Real number-based code
- Integer/symbol-based code
- Permutation-based code
- Data structure-based code

For the functional / constrained optimization problem, the real number-based code are widely used, and confirmed that the performance is higher than the binary string-based code. For the combinatorial optimization problems, the integer/symbol-base code and the permutation-based code are widely used. For the practical problems, the data structure-based code is easy to capture the essence of the problem. According the structure of the data structure-based code, it can be divided into two types:

- one-dimensional structure code
- multi-dimensional structure code

The most EAs consider the one-dimensional code. However, there are many practical problems considered with multiple dimensional structure. Naturally, adopt the multi-dimensional code to represent the solution. With considering the implementation of encoding and decoding, the representations also can included the following information:

- only solution
- solution and implementation parameters

Traditional EDA

Estimation of distribution algorithm (EDA) is stochastic optimization methods, guiding by establishing and search the optimal sampling probability model of candidate solutions. EDA as a model-based optimization approach can solve many kinds of complex optimization problems. Fig.2.4 shows an optimization process of illustration of EDA approaches. EDA processes as the following steps:

S.1 the initial population $pop(0)$ of Q individuals is generated. We usually assume a distribution of variables to generate Q individuals, and next each individual v_j in $Pop(0)$ is evaluated.

S.2 in order to make the t-th population $pop(t)$ evolve towards the next $pop(t+1)$ one, a number $S(S < Q)$ of individuals are selected from $pop(t)$ following a criterion. We denote by D the set of S selected individuals from generation t.

S.3 induce the n-dimensional probabilistic model $M(t)$ that better represents the interdependencies between the n variables. This step is also known as the learning procedure, and this step is also called the learning process, which is the most critical, as the appropriate dependencies between variables on the bench is crucial to the development of appropriate individuals.

Steps 2, 3 and 4 are repeated until it reaches the end condition. Given a file of cases and Fig.2.5 shows the general implementation process of EDA.

Classification of EDAs

In discrete and/or continuous regions, the optimization and probabilistic modeling have different nature. How to develop an effective EDA approach for each of regions, depend on the representation type with different problem types. In this sub-section, we introduce some different probabilistic models and structure learning approaches employed in EDAs. Generally, to develop an effective EDA approach, we have to consider (1) what kind of probabilistic model should be used, which can accurately represent the characteristics of the

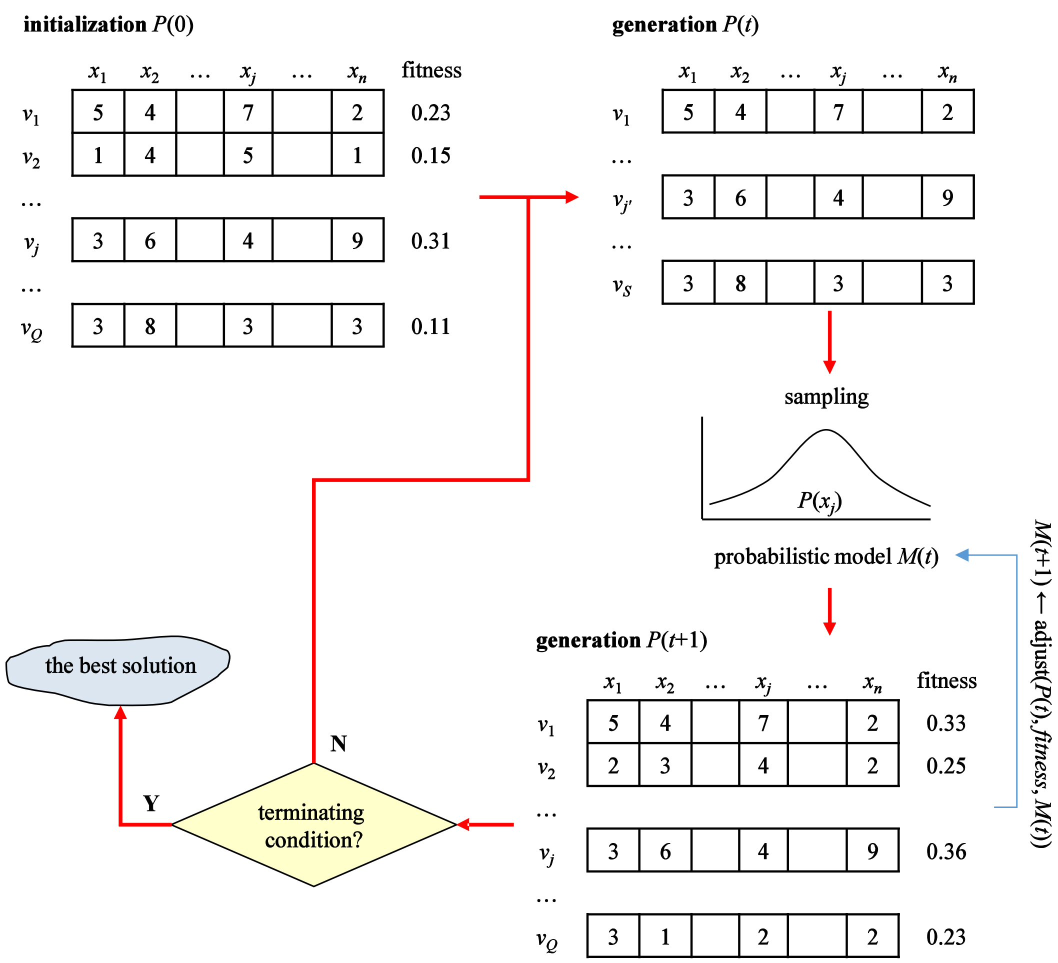


Fig. 2.4 Illustration of EDA in optimization process

problem, and (2) the complexity of the probabilistic models can be accepted. We summarized the probabilistic models adopted into EDAs, listed the algorithm names in Table 2.1.

Univariate EDA

As shown in Fig.2.6, the decision variables of the problem are independent in univariate EDA. The probability distribution of one decision variable is not depend with any other decision variables. The probability of single variable model decomposition of candidate solutions $(x_1, x_2, \ldots, x_n)$ product of the probability of a single variable:

$$p(x_1, x_2, \ldots, x_n) = p(x_1)p(x_2)\ldots p(x_n) \tag{2.1}$$

General Estimation of Distribution Algorithm (EDA)

Define:
 $X = (X_1, X_2, \ldots, X_n)$: vector of the decision variables;
 Q: number of the population solutions kept by EDA;
 S: a number $S(S < Q)$ of individuals are selected from $Pop(t)$;
end
Input: problem data, parameters
Output: s: the best solution

---

begin
 Initialization
  $t \leftarrow 0$;
  Initialize the probabilistic model $M(t)$ with prior probability;
 end
 while *not terminating condition* **do**
  step 1: Sample a set of new solutions $pop(t)$ conditioned on $M(t)$;
  step 2: Evaluate $pop(t)$, and update the best solution s;
  step 3: Select the promising data set D from $pop(t)$ consisting of Q solutions;
  step 4: Adjust the probabilistic model $M(t)$ according to D ;
  step 5: $t \leftarrow t + 1$;
 end
 Output: return the best solution s;
end

Fig. 2.5 Pseudo-code of general Estimation of Distribution Algorithm

where $p(x_i)$ is the probability of variable x_i. $p(x_1, x_2, \ldots, x_n)$ is the probability of candidate solutions $(x_1, x_2, \ldots, x_n)$. The single variable model by n variable probability tables. Each table defines the probability of the different values of corresponding variable, which the sumation of the probability must to be 1.

MIMIC

Mutual information maximizing input clustering (MIMIC) (shown in Fig.2.7): MIMIC-based EDAs use the bivariate models, extend the capability of EDAs' modeling. By using efficient learning methods, bivariate models represent the pair-dependencies between EDAs' variables. MIMC used the probability model of the chain structure the probability distribution of all variables, in addition to the first node condition before the variable's value chain [18]. Given a permutation of the n variables in a problem, $\pi = i_1, i_2, \ldots i_n$, we can show the probability distribution of $p(x_1, x_2, \ldots, x_n)$ by MIMIC decomposes as:

Table 2.1 EDAs with Graphical Models

Algorithms	Full name	Probabilistic Model
UEDA	Univariate EDA	Independent
MIMIC	Mutual Information Maximizing Input Clustering	Chain
COMIT	Combining Optimizers with Mutual Information Tress	Tree
BMDA	Bivariate Marginal Distribution Algorithm	Forest
ECGA	Extended Compact Genetic Algorithm	Marginal Product Model
BOA	Bayesian Optimization Algorithm	Bayesian Network
MNEDA	Markov Network based EDA	Markov Network

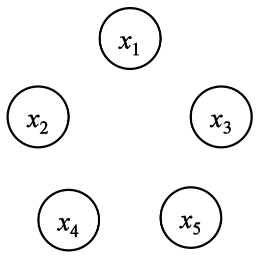


Fig. 2.6 Independent examples of probabilistic model

$$p_\pi(x_1, x_2, \ldots, x_n) = p(x_{i1}|x_{i2})p(x_{i2}|x_{i3})\ldots p(x_{i,n-1}|x_{in})p(x_{in}) \tag{2.2}$$

where $p(x_{ij}|x_{i,j+1})$ means the conditional probability of x_{ij} with $x_{i,j+1}$. Then generate new candidate solutions through the probability distribution model of sampling coding. Generate income variable sampling arranged in reverse chronological order. This order respect the permutation π, and start from x_{in} with end of x_{i1}.

COMIT

Combining optimizers with mutual information tress (COMIT) (shown in Fig.2.8): Baluja and Davis (1997) promising solutions use dependency tree model, the probability model to improve performance, compared to the chain model to simulate [19]. In a dependency tree, each parents can have more than one child.The increment of EDA is by using a probability

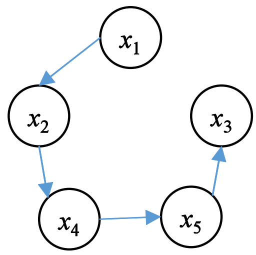


Fig. 2.7 Chain examples of probabilistic model

matrix contains all pairs of probability. Model is the maximum mutual information connection between tree variables, namely the prove to best tree model Kullback Leibler - divergence on the true distribution [20]. For all candidate solutions, COMIT initialize the probability matrix, corresponding to the uniform distribution. In each iteration, tree model was building and sampling to generate new solutions. Then the best one of update probability matrix using these solutions.

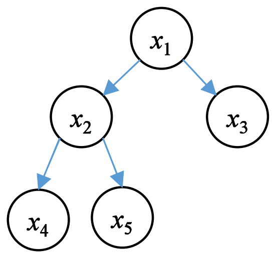


Fig. 2.8 Tree examples of probabilistic model

BMDA

Bivariate marginal distribution algorithm (BMDA) (shown in Fig.2.9): A set of mutually independent trees is used to create a probabilistic model [21]. As the main measure rules of dependence by using Pearson's chi-square statistics [22], in each generation, a dependency model can be created. Then, sampling to produce a new solution model based on conditional probability from the population. Extended compact genetic algorithm (ECGA) (shown in Fig.2.10) uees a model that could be divided into independent variables cluster, each cluster is treated [23]. The establishment of the model first assume that all variables are the issue of independence. The model quality is used to measure the minimum description length (MDL), which two cluster together improve the quality of the model in each iteration.

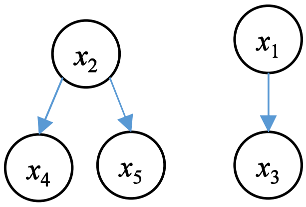


Fig. 2.9 Forest examples of probabilistic model

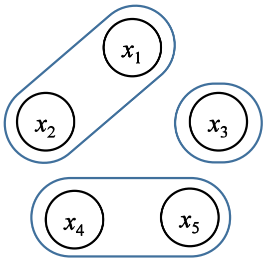


Fig. 2.10 ECGA model

BOA

Bayesian optimization algorithm (BOA) (shown in Fig.2.11): Many problems meets the same challenge that the problem cannot be accurately described by a model with dividing the problem to independent sub-problems. BOA candidate solutions uses Bayesian network model, which makes the problems can be solved nearly decomposable categories [24]. Bayesian network is a kind of acyclic directed graph (ADG), has a each variable node, the node between side expressed conditional dependencies.

in Bayesian network, there are n nodes coding of the joint probability distribution, with considering the random variables $(x_1, x_2, \ldots, x_n)$:

$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i|P_i) \tag{2.3}$$

where x_i is the members of parents P_i, a set of variables. $p(x_i|P_i)$ is the conditional probability of x_i with P_i gave. Bayesian network and tree model in bayesian networks, is the difference between each variable may depend on multiple variables. Bayesian network is a Bayesian network can capture more complicated problem decomposition of subproblems interactions.

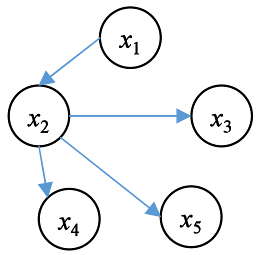


Fig. 2.11 Bayesian network for probabilistic model

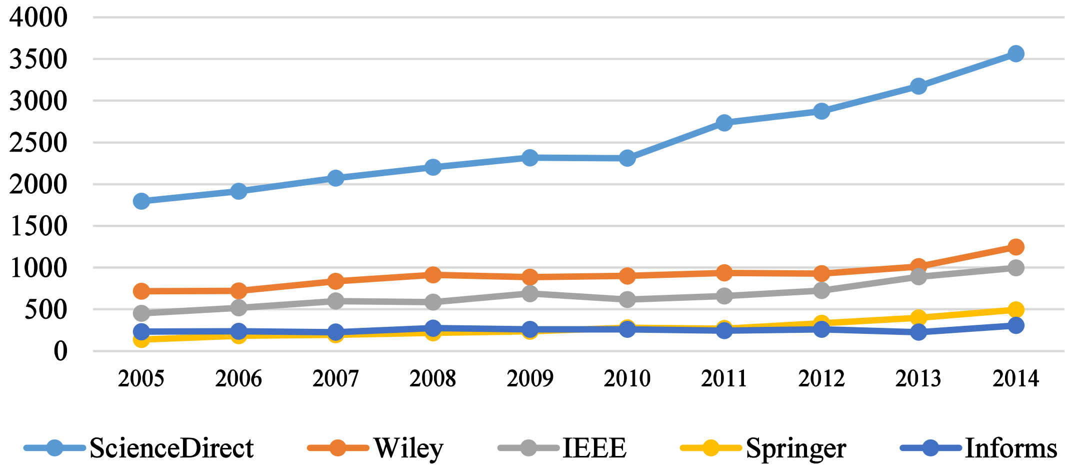


Fig. 2.12 Number of papers published on scheduling

However, in the research, Greedy strategy based heuristic approach is employed to learning the structure of Bayesian network, computation is taken more costs has been applied to solve numerous applications, particularly for the large-scale problem.

2.1.2 Application of EDA to Scheduling

These features significantly increase the complexity of finding ever approximately optimal solutions. Therefore, the scheduling problems have abstracts more attention both in academia and in industries. Fig.2.12 shows that the number of papers related scheduling grow at an average annual rate of 8% in this decade. Particularly, In 2014, there are more than 6600 papers published, rise by nearly 15% over the same period in 2013.

In job shop agile manufacturing environment, the working area of the machinery, cell , consist of multiple-purpose machines to deal with the product-mixed model. The processing of tasks in cell can be organized flexible job shop manner. Therefore, we can treat the scheduling problems in AMS as a sort of complex flexible job scheduling problem (FJSP) . FJSP is described as n operations need to be performed on m machines, the processing time varies according to the multiple-purpose machines. The decision-making space can be considered as (1) how to perform the operations by an operation sequence (tasks permutation); (2) how to allocate the resources to each operation (machine allocation).

The combinatorial optimization problem considered here is: maximum (or minimum) a non-linear, but deterministic function $f(\boldsymbol{x})$ which depends on multiple dimensional variables $\boldsymbol{x}$, is shown as follows:

$$f(\boldsymbol{x}^*) = \min_{x\in\Omega} f(\boldsymbol{x}) \tag{2.4}$$

where $\boldsymbol{x} = \{x_1, \ldots, x_n\}$ is a n-dimensional vector, also called an individual, with domain $\Omega \in R_n$ In most cases, the function $f(\boldsymbol{x})$ spends expensively to evaluate, and cannot be solved efficiently and/or accurately. The high dimensionality of variables $\boldsymbol{x}$ make the computational requirements more excessive for searching the optimal solution and it is referred to "curse of dimension". Combinatorial optimization problem with $n > 100$ is always called high-dimensional optimization problems .

Furthermore, FJSP can be formulated by $f(\boldsymbol{x}^*, \boldsymbol{y}^*)$, defined in the following:

$$f(\boldsymbol{x}^*, \boldsymbol{y}^*) = \min_{x\in\Omega} f(\boldsymbol{x}, \boldsymbol{y}) \tag{2.5}$$

where $\boldsymbol{x}$ is a n-dimensional vector, $\boldsymbol{x} = \{x_1, \ldots, x_n\}$, represented the precedence sequence of n operations, and $\boldsymbol{y}$ is a n-dimensional vector, $\boldsymbol{y} = \{y_1, \ldots, y_n\}$, represented machine allocation of n operations.

The operation permutation determines which operation can select the resource firstly. The performance of scheduling is affected by resources allocation if the operations permutation was determined.

Currently, the process of scheduling problem is from simple to complex, the processing characteristics is from static to dynamic and the evaluation criteria is from the single target to multiple targets. With the scheduling modeling become more diversified and complex, the research methods have changed from mathematical methods to intelligent heuristic algorithms along with the change of scheduling modeling. Li and Ierapetritou pointed out that although most of the scheduling problems are regarded as NP hard problem, they are still less uncomplicated than the complexity of practical problems [25].

Definition: Let A be an algorithm with the set $\boldsymbol{x}$ as its inputs, and let $f : R^n \to R_+$, if there exists the constants $\alpha, \beta > 0$ make the algorithm compete in $\alpha f(\text{size}(x)) + \beta$ elementary steps (including arithmetic operations) for each $x \in \boldsymbol{x}$, we say A runs in $O(f)$ time or the running time of A is $O(f)$.

The processing time is different according to the instances although they have the same size. We consider the worst case.The function $f : R^n \to R$, where $f(n)$ represents the maximum processing time of an instance which has n dimension. The worst-case is a pessimistic measure.

Let A be an algorithm, each input $x \in R^n$ computes $f(x) \in R$, we call A computes $f : R^n \to R$. If f can be computed in polynomial-time by some algorithms A, we call the algorithms "good algorithms" or "efficient algorithms". Table2.2 shows hypothetical running times of various time complexities algorithms [26]. For different dimension n, one compute step takes 1 nanosecond, the running of algorithms take $100n\log n, 10n^2, n^{3.5}, n^{\log n}, 2^n, n!$ elementary steps.

In job shop problems, for operations sequence, the running time of algorithm take $n!$ elementary steps, and for resources allocation, the running time of algorithm take nm elementary steps. Because of the interdependence of the resources allocation and operations permutation, the running time of FJSP algorithms take $(nm)!$ elementary steps.

Definition: Let n be an input size and k be an integer number, and the intermediate computations of an algorithm can be stored with running time $O(n^k)$. If it runs in $O(n^k)$ for any input consisting of n numbers, the algorithm run in strongly polynomial time. If $k = 1$, the algorithm called a linear-time algorithm. If it runs in polynomial but not strongly polynomial time, the algorithm called weakly polynomial.

As shown in Table2.2, although polynomial-time algorithm is faster for large enough instances, when considering the constant exponent of input size n. Unfortunately, not all problem are not amenable to the polynomial-time algorithm, and a broad class of highly important problems fall into a category for which polynomial-time algorithms are extremely unlikely to exist. The running time of algorithms that take $n\log n$, 2^n, $n!$ elementary steps, can be unacceptable when the times of compute step are exponential growth. For decades, the researchers have unsuccessfully tried to find efficient algorithms for above class contains many problems. The most such class of above problems are NP-complete problems. To find an efficient (polynomial time) method would therefore give rise to efficient algorithms for all NP-complete problems, an extremely unlikely event. In other words, by showing that a problem is NP-complete, we are essentially showing that it is extremely unlikely to have an efficient solution. Therefore, solving the FJSP problems (the running time $O((nm)!)$) is very difficult. So, how to increase the effectiveness of scheduling methods in effective time is still one of the important research in scheduling problem.

Table 2.2 Hypothetical running times of algorithms with various time complexities

n	$100n^{\log n}$	$10n^2$	$n^{3.5}$	$n^{\log n}$	2^n	$n!$
10	3µs	1µs	3µs	2µs	1µs	4ms
20	9µs	4µs	36µs	420µs	1ms	76 years
30	15µs	9µs	148µs	20ms	1s	$8x10^{15}$ y
40	21µs	16µs	404µs	340ms	1100s	
50	28µs	25µs	884µs	4s	13 days	
60	35µs	36µs	2ms	32s	37 years	
80	50µs	64µs	5ms	1075s	$4x10^7$ y	
100	66µs	100µs	10ms	5 hours	$4x10^{13}$ y	
200	153µs	400µs	113ms	12 years		
500	448µs	2.5ms	3s	$5x10^5$ y		
1000	1ms	10ms	32s	$3x10^{13}$ y		
10^4	13ms	1s	28 hours			
10^5	166ms	100s	10 years			
10^6	2s	3 hours	3169 years			
10^7	23s	12 days	10^7 y			
10^8	266s	3 years	$3x10^{10}$ y			
10^{10}	9 hours	$3x10^4$ y				
10^{12}	46 days	$3x10^8$ y				

Scheduling Methodologies

Currently, the process of scheduling problem is from simple to complex, the processing characteristics is from static to dynamic and the evaluation criteria is from the single target to multiple targets. With the scheduling modeling become more diversified and complex, the research methods have changed from mathematical methods to intelligent heuristic algorithms along with the change of scheduling modeling. Li and Ierapetritou pointed out that although most of the scheduling problems are regarded as NP hard problem, they are still less uncomplicated than the complexity of practical problems [25]. So, how to increase the effectiveness of scheduling methods in effective time is still one of the important research in scheduling problem. this section review the scheduling methodologies.

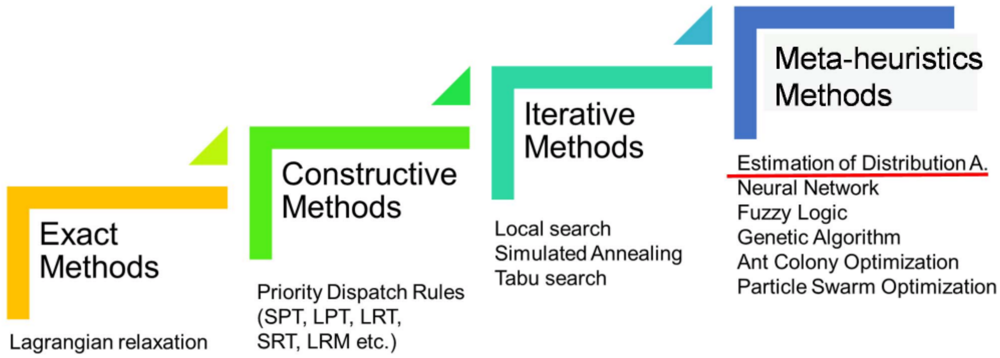


Fig. 2.13 Survey of scheduling methodologies

a) Exact Methods

Among the methods used for scheduling, the relatively successful mathematical modeling is Lagrangian relaxation. Luh from University of Connecticut made a systematically research of the Lagrangian relaxation [27, 28], decomposed scheduling into two parts: task-based decomposition and operation-based decomposition. On the basis of exist studies, Hoitomt et.al (1993) first used the augmentation Lagrangian Relaxation for solving scheduling, and got better solutions [29]. Due to the scheduling problem need to balance the operation time and the resource selection, when it decomposed by Lagrangian relaxation, it may cause unrest and make the dual problem with slow converge or no converge [11]. In recent years, most of researchers solve the scheduling problem by combining Lagrangian Relaxation and other optimization methods, such as combining with dynamic programming method [30, 31], using the Artificial Neural Network to optimize the sub-problem decomposition [32] and so on. Due to the complex of the scheduling problem, the computing time of most accurate methods grows exponentially over the scale of problem.

b) Constructive Methods

Priority Dispatch Rules (PDR) is the earliest approximation method[Smith56], the main idea is assigning priority for each operation, select the operation with biggest priority for each time to form the operation sequence. Panwalkar and Iskander summarized and concluded 113 rules of scheduling and classified them [33]. In practice, the most common method have SPT, LPT, LRT, SRT, LRM and so on [34]. Recently, researchers solve the scheduling problem through combining the priority assign rules with other optimization methods [35, 36]. But, the kind of approaches have the weakness of only considering the current status of resources and the effectiveness of solutions.

Shifting Bottleneck Procedure (SBP) is one of the most effective constructor method for solving scheduling problem. It is proposed by Adams et.al in 1998, it is the first heuristic algorithms for FT10 standard test examples [37]. In recent years, researchers find that combing the Shifting Bottleneck procedure with iterative algorithm can get more effective solutions. Mönch et.al (2007) proposed a heuristic algorithm which was based Shifting Bottleneck procedure for complex scheduling problems [38]. Gao et.al (2007) proposed a hybrid genetic algorithm which was based on SBP for multiple-objective scheduling problem [39]. Cheng et.al (2011) combined SBP, priority assign rules and evolutionary algorithms for solving multi-objective scheduling problem [40].

c) Iterative Methods

Local search evaluated from early heuristic algorithms, which is represented by Simulated Annealing and Tabu Search. In 1983, Kirkpatrick et.al proposed the Simulated Annealing (SA) [41], it evaluated from the combination of physical process of solid annealing and Metropolis rules [42]. Koulamas et.al (1994) gave the literature review of using SA to solve the scheduling problems [43]. Loukil et.al (2007) proposed a SA method which considered multi-objective scheduling problem [44]. Golver and Hansen (1996) proposed the Tuba search [45, 46], during the runtime, Tuba search produces an initial solution by a certain method, then ,searches all feasible solutions within the neighborhood ,final, chooses the best solution as the current solution. Bożejko (2010) proposed a parallel hybrid heuristic method for scheduling problems, which combined Tabu search and population-based optimization method [47]. Li et.al (2010) proposed a hybrid Tabu search which combined the Tabu search and local search for multi-objective scheduling problems [48]. Zhang et.al (2012) proposed a hybrid method with Tabu search and genetic algorithm which used to study the resource and processing time of scheduling [49]. The advantage of local search is that it can find out the local best solution in shorter time, but is dependent of structural features and initial solutions of problems, that makes it easily to local optimum but can't get the global optimum. Recently, researchers combined the local search and the population-based optimization method to improve the solution and computing time of scheduling problem.

In 1980s, Artificial Intelligence (AI) came into researchers' view, it played an important role in scheduling, and offered an effective method for solving scheduling. Expert System (ES) is a computer program which can offer human mind to solve complex problem in certain areas. There are some famous ES, such as: ISIS [50], SOJA [51], OPIS [52], CORTES [53] and so on. But ES is only used for certain areas because of its expensive cost of development, its bad adaptability to the environment and the large scale of accumulation of experience and knowledge of scheduling. Multi Agent System (MAS) is a composition of one or more multi agent, it makes large and complex system into some sub-system which can communicate and coordinate to each other and can be easily managed. MAS is widely used in real dynamic system because it has the advantage of well flexibility and adaptability to the dynamic environment. For research about scheduling based on MAS, Shen et.al (2006) gave the literature review about scheduling method based on MAS [54]. They classified scheduling methods into object-oriented methods and agent-based methods and in the future work, they emphasized multi-agent need be do deeper study and combining with other targeted algorithm is necessary.

d) Evolutionary Computation

Evolutionary computation (EC) got the wide attention of researchers because of its intelligent, parallelism, robustness and well adaptability and the capability of global search. The main methods of EC are Genetic Algorithm (GA), Genetic Algorithm (GA), Neural Network (NN), Ant Colony Optimization (ACO) and etc. Holland (1975) published the first influential book about GA named Adaptation in Natural and Artificial System. In 1985, Davis first used GA for scheduling problem, proved the effectiveness of GA by the numerical experiments with 20x6 Shop scheduling problems [55]. Nakano et.al (1991) used GA for various scheduling problems. In 1990s, the applied research based on computational intelligence got widely attention in the scheduling problem. Recently, there are lots of literature about using computational intelligence for Shop scheduling problems [56, 57]. Meanwhile, researchers did several review about the scheduling optimization algorithms in different periods [58–60].

Although the competing intelligence methods are diversity, most of them have same set of features: the process of finding solutions is iterative-based process, use the population-based searching mechanism, great attention to evaluation systems and fitness-based evolution and parameter-based control mechanism. So most design of algorithm focus on improvements on the above several mechanisms [61]. Lin and Gen (2009) designed a set of fuzzy logic control mechanism, in order to realize the parameters of the dynamic adjustment in the process of iteration algorithm, namely improved the parameter control mechanism [62]. Michalewicz et.al (1994) pointed out that most of the computational intelligence method must be based on the structure characteristics of the problem [17].When design the algorithm according to the structure characteristics of the problem, most researchers pay more attention to evaluation systems and fitness-based evolution of solution.

Table 2.3 Survey of the Application of EDA in Scheduling

Applications	References	Criterion/environment	Approach/Comments
flow-shop scheduling problem	Jarboui and Eddaly (2009)	total flowtime	DEDA (Marginal Distribution)
	Pan and Ruiz (2012)	total flowtime |setep Time	DEDA (Marginal Distribution)
	Liu and wang (2013)	makespan|uncertainty	DEDA (Marginal Distribution)
	Ceberio *et al*. (2014)	makespan	DEDA(Gaussian probability distribution)
	Want, Choi and Lu (2015)	makespan|uncertainty	Simulation-based BEDA
	Liu *et al.* (2015)	makespan|uncertainty	OCBA-based DEDA (enhanced Learning)
job-shop scheduling	Zhang (2011)	total weighted tardiness	DEDA (Marginal Distribution of dispatching rules)
	He *et al.* (2013)	makespan	DEDA (Marginal Distribution)
	Zhao *et al.* (2015)	makespan|uncertainty	Neighborhood-based EDA
Flexible job-shop scheduling	Wang *et al*. (2013)	makespan	BEDA
	Pérez-Rodríguez *et al.* (2014)	makespan	Simulation-based BEDA
Laborer related scheduling	Aickelin and Li (2007) (*nurse scheduling problem*)	complete Time	Naive Bayes-based EDA
	Wang and Chen (2012) (*project scheduling problem*)	complete Time	SGS-based HEDA
	Fang et al. (2015) (*stochastic project scheduling problem*)	complete Time	PBLS-based EDA

The scheduling problem can be decomposed as two parts: resource selection and operation sequence. In general, there are two different model: integration model and distributed model. Integration model represents considering sub-problem at one time; sequential model shows to consider the sub-problem one after another. But integration model is easy to trap in local optimum. At present, most of the researchers focus on designing the scheduling algorithm based on the integration model. Therefore, in scheduling problem, most of the research focus on the validity of algorithm, how to effectively decompose problem into sub-problem is token relatively few into consideration. Transferring the algorithm design of problem oriented features to characteristics of the problem analysis has certain feasibility.

Survey of the Application of EDA in Scheduling

In the previous section, we give the overview of theories about EDAs. This section is to provide comprehensive review of the application of EDAs on the scheduling problems according classification of the environment.

Flow-Shop Problem (FSP)

Jarboui and Eddaly (2009) proposed discrete EDA aiming at minimizing the total flowtime in permutation flow-shop scheduling problems [63]. In this study, the probabilistic model built focuses on both the importance of the order of the jobs in the sequences and the similar blocks of jobs presented in the selected parents.

Pan and Ruiz (2012) considered ledot-streaming flow shop scheduling problem with sequence-dependent setup times under both the idling and no-idling production cases [64]. The studied objective is makespan minimization. In the author's approach, Jarboui's proposed probabilistic model is adopted in EDA. Furthermore, some advanced techniques such as initialization of population and diversity controlling mechanism are incorporated to achieve the best optimality.

Liu and wang (2013) first studied proposed for solving the distributed permutation flow-shop scheduling problem (DPFSP) with the criterion to minimize the makespan [65]. The studied objective is makespan minimization. The experiments shows that the EDA is more effective than the existing methods in solving the DPFSP.

Ceberio *et al.* (2014) developed generalized Mallows model based EDA to to deal with permutation-based optimization problems. Different to the previous approaches. Mallows model is distance based exponential probabilistic model considered analogous to the Gaussian probability distribution over the space of permutations [66].

Want, Choi and Lu (2015) presented a two-stage simulation-based hybrid estimation of distribution algorithm (TSSB-HEDA) to schedule the permutation flow-shop under stochastic processing times. [67]. Moreover, a self-adaptive learning mechanism (SALM) is employed to dynamically adjust the ratio of offspring individuals generated by the probabilistic model.

Want, Wang, Liu and Xu (2015) also addressed the scheduling problems with uncertainty [68]. In this study, optimal computing budget allocation (OCBA) technique is developed to provide a reliable identification to the good solutions among the population. Consequently, the superior individuals identified by the OCBA can update the probability model more efficiently. The numerical testing results and comparisons with the existing algorithm are provided, which demonstrate the effectiveness of the proposed OEDA.

From the previous literature, The probabilistic model encoding importance of the order of the jobs in the sequences and the similar blocks of jobs presented in the selected parents is widely adopted to model the permutation problem in FSP by researchers.

Job-Shop Problem (JSP)

Zhang (2011) considered job shop scheduling problems with the total weighted tardiness objective [69]. several effective dispatching rules based heuristic is employed to construct feasible schedule. In order to achieve better performance, EDA is proposed to learn the combination of the rules.

He *et al*. (2013) developed new estimation of distribution algorithm for job shop scheduling problems based on Bayesian statistical inference theory. [70]. According to permutations of all operations on machines, the model of priori distribution probability is built by extracting information of superior solutions updating, then the model of conditional probability is also built based on the frequencies of neighboring operations appearing. The model of posterior probability combining the above two models with Bayesian formula is used to guide new populations generation.

Zhao *et al*. (2015) also addressed the scheduling problems with uncertainty [71]. In this study, optimal computing budget allocation (OCBA) technique is developed to provide a reliable identification to the good solutions among the population. Consequently, the superior individuals identified by the OCBA can update the probability model more efficiently. The numerical testing results and comparisons with the existing algorithm are provided, which demonstrate the effectiveness of the proposed OEDA.

From the optimization view, JSP could be thought as permutation problem where the processing routing of job is determined. Ideally, the probabilistic model in FSP is suitable for permutation problem in JSP.

Flexible Job-Shop Problem (FJSP)

Wang *et al*. (2012) developed An effective EDA for solving the flexible job-shop scheduling problem (BEDA) [72]. In the proposal, operation probability matrix is designed to model the distribution of importance of operation in operation sequence, and machine probability matrix represents the probability of machine selection per operation. Moreover, a left-shift scheme is employed for improving schedule solution. Wang and Liu (2013) extended the previous approach for solving the flexible job-shop scheduling problem with fuzzy processing time in real-world scheduling [73].

Pérez-Rodríguez *et al*. (2014) study simulation optimization using an estimation of distribution algorithm for solving the flexible job-shop scheduling problem [74]. In the proposal, three different models are taken into account. The first tree model aims to model the distribution of operation sequence by continuous domains. The second probabilistic model is similar to Wang's approach. The final probabilistic model aims to learn the distribution of off-duty hours work shift.

Laborer related Scheduling

Aickelin and Li (2007) this paper presented a novel EDA for the nurse scheduling problem, which involves choosing a suitable scheduling rule from a set for the assignment of each nurse [75]. In the proposal, Bayesian network is to construct the joint distribution of the rule selection per nurse.

Wang and Chen (2012) this paper developed a hybrid estimation of distribution algorithm (HEDA) to solve the resource-constrained project scheduling problem (RCPSP) [76]. In the HEDA, the individuals are encoded based on the extended active list (EAL) and decoded by serial schedule generation scheme (SGS). The probability model updating mechanism is designed for well sampling the promising searching region.

Fang *et al.* (2015) this paper designed an effective EDA to solve the stochastic resource-constrained project scheduling problem [77]. In this study, the permutation-based local search strategy (PBLS) is proposed to enhance the exploitation ability. Simulation results based on the PSPLIB benchmarks and comparisons with some existing algorithms demonstrated the effectiveness of the proposed EDA.

2.2 Machine learning

Machine learning is prediction techniques of data analysis that automates analytical model building. In learning community, one of important task is to reason probabilistically about the values of one or more of the variables, possibly given observations about some others. In order to do so using principled probabilistic reasoning, we need to construct a joint distribution over the space of possible assignments to distribution some set of random variables. In this scenario, the structure of model encoding the dependency between the variables play an important role in inference. Therefore, in this section, we review two representation concerning probabilistic graphical model (PGM),Bayesian network (BS) and Markov random field (MR) and the related application.

2.2.1 Bayesian network

Bayesian networks build on the intuitions by exploiting independent property distribution to allow a compact and natural said. However, they are not limited to represent the distribution meet the strong independence hypothesis implied.They allow us the flexibility to tailor our representation of the distribution to the independence properties that appear reasonable in the current settings.

The soul of the Bayesian network is that the structure is a Directed Acyclic Graph (DAG) G , whose edges of the nodes domain and the corresponding random variables, intuitively, to influence of one node on another nodes directly. This graph G can be viewed in two very different ways:

(a) As the data structure, it provides the framework for representing a compact joint distribution in a factorized way.

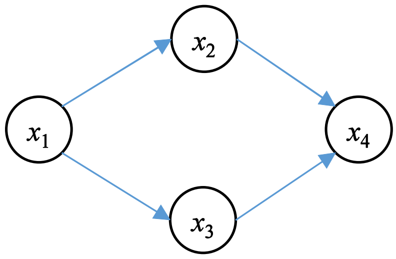
(b) As the representation, it shows a set of conditional independence assumption.

The second component of the Bayesian network representation is a set of local probability model models that represent each variable dependencies in the nature of their parents. In general, each variable X of CPD model and the conditional probability distribution (CPD) , specify a model on the value of X given every possible combination of values assigned to their parents.For a node has no parents, CPD is an empty set of condition variables.Therefore, CPD changes into marginal distribution. Pelikan introduced Bayesian optimization algorithm (BOA) [78], and the variables are estimated based on the Bayesian network model data joint distribution. Each node represent one variable, and the directed edge between two nodes represents the relationship of corresponding nodes in the network. There exists he probabilistic relationships among the data which used for modeling the network structure, the relationships can be represented by one Bayesian network. The got networks shows the structure of the problems. The mathematical representation of the Bayesian network can defined as follows:

$$p(X) = \prod_{i=1}^{n} p(X_i|\Pi_{X_i}) \tag{2.6}$$

where $X = (X_1, \ldots, X_n)$ is a vector, $\Pi(X_i)$ denotes the variables set of the problems, and $p(X_i|\Pi_{X_i})$ is the conditional probability of X_i conditioned under ΠX_i. Fig.2.14 shows a simple of Bayesian network with 4 variables, the graphical model presents the relation between the variables.

According to representation of BS, Learning Bayesian network consists of two functions: the first function is Bayesian network Learning Structure , which build Parenthoods system to model encode causal –relationship; the other function is parameter Estimation which estimate conditional probability table of each variable.



$$P(X) = P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|X_1,X_3)$$

Fig. 2.14 Illustrative sample of Bayesian network and the encoded joint distribution

2.2.2 Markov random field

As discussed above, we have dealt only with directed graphical models, namely Bayesian networks. These models are useful because both the structure and the parameters provide a natural representation for many types of real-world domains. In the following, we turn our attention to another important class of graphical models, defined on the basis of undirected graphs. Then, we will see, these models helps to model the phenomena which the interaction between variables could not be described directionally. Furthermore, the undirected models offer both the independence structure and the inference tasks a different model, which it is often simpler perspective on directed models. We also introduce a combined framework that allows both directed and undirected edges.

There exists a scene that to use the undirected graph to represent this intuition.The nodes represent the variables which is the same as the definition in Bayesian network. The arcs represents the direct probabilistic interaction between the linked variables — an interaction which is not decided by any variables in the network. The next question is determine the parameters of the undirected graph. Because the interaction is not directed, there is no reason to use a standard CPD, where we represent the distribution over one node given others. Rather, we need a more symmetric parameterization. Intuitively, what we want to capture is the affinities between related variables. For example, we might want to represent the fact that Alice and Bob are more likely to choose "agree" than to choose "disagree". We with the general function of the A and B, also called a factor. As in a Bayesian network, the parameterization of Markov network defines the local interactions between directly related variables.

The Markov network can be represented by (G, Ψ), in which G represents the structure of network and Ψ represents parameters set. Markov network is an undirected graph, in which the nodes represents variables which belongs to the data set used to by modeled and each arc represents the conditional dependency between linked nodes. However, there are some difference between Markov network and Bayesian network, that is, the arcs in Markov network are undirected, so compared to the causal relationship, to treat as the neighborhood relationship between two linked nodes is much better. We use N to represents the neighborhood relationship set of G, in which each N_i is the set of neighborhood nodes of node X_i. For discrete decision values, the domain of X_i represents the value set of $D(X_i) = \{x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n_i)}\}$. Fig.2.15 shown an simple example of Markov network structure which has 5 random variables. As shown, X_1 has 2 neighbors $N_1 = \{X_2, X_3\}$,X_2 has 3 neighbors $N_2 = \{X_1, X_3, X_4\}$ and X_4 has 2 neighbors $N_2 = \{X_2, X_5\}$.

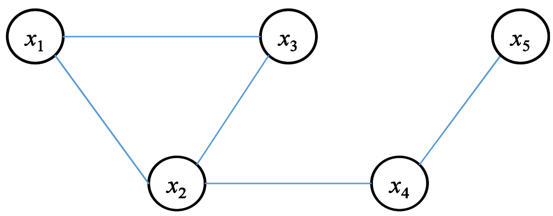


Fig. 2.15 A simple Markov network with 5 random variables

The Markov network is modeled based on the local Markov property among the neighborhood relationship of variables. The local Markov property means when given the neighborhoods of one variable, it will be conditional independence of other variables. It means that, the given conditional probability of the neighborhood N_i of this node can compute the conditional probability of a node X_i.It can be written as follows according to the probability:

$$p(x_i|x - \{x_i\}) = p(x_i|N_i) \tag{2.7}$$

As it is difficult to establish the Markov properties with an arbitrary probability distribution, the common approach of Markov random fields is to factorize based on the cliques of the graph. The joint probability distribution can be represented as

$$p(x) = \frac{1}{Z} \prod_{l=1}^{m} \Psi_i(c_i) \tag{2.8}$$

where, $\Psi_i(c_i)$ on behalf of a clique of potential function $c_i \in X$, and m represents the number of cliques in the structure G. The normalizing constant $Z = \sum_{x\in\Omega} \prod_{i-1}^{m} \Psi_i(c_i)$ and Z is the *partition function* with mensuration of $\sum_{x\in\Omega} p(x) = 1$.

Similar to Bayesian network, Learning Markov Random Field consists of two functions: the first function is learning Structure , which build Neighborhoods system to model encode the interaction of the variables. The second function is Parameter Estimation which estimate conditional probability table of each variable.

The main difference between them is that they use different types of diagram to express the relationship between the variables: BS use the Directed Acyclic Graph (DAG) to express causality; MR use the undirected graph to express the interactions between variables. The different structure makes them have a series of subtle differences in modeling and inference. In general, each node in a Bayesian Network corresponds to a prior probability distribution or conditional probability distribution, so the joint distribution of the whole structure can be directly decomposed into the product of the distribution for each single node. Nevertheless, because of no specific causality, the joint probability distribution is often described as the product of several potential functions. However, we need to make the joint probability distribution to be 1 to make it effective, this causes great difficulties to parameter estimation in the actual application.

On the other hand, we provide comprehensive review of EDA-related approaches for solving scheduling problem. It is given that so many papers have appeared in a short time, most of researches well study flow-shop based scheduling problems where the probabilistic model focuses on both the importance of the order of the jobs in the sequences, and the similar blocks of jobs presented in the selected parents. This idea is well referred in the recent researches. Although few researches presented novel EDAs for shop-based scheduling and flexible job scheduling problem. However, to the best of our knowledge, there is no research work about the EDA considering interaction among the machine allocation during scheduling process. So, in order to model interaction among the variables , the Probabilistic Graphical Models-based learning technique should be incorporated into EDAs.

Chapter 3

Problem description and Approach for solving problems

In this chapter, we present the main idea of our proposed novel meta-heuristic algorithm of Hybridized Estimation of Distribution Algorithm with Probabilistic Graphical Models (GPM). Firstly, We present interdependencies between operations during machine allocation is described in detail. This interdependency significantly affect the performance of scheduling problem optimization considering the criteria such as make-span.

Next, the main framework of our proposal and principal components are given respectively. It should be noted that machine and auxiliary resource such as the tools equipped with machine is collectively referred to *resource* in the section, if it is not being misunderstood.

3.1 Problem description

In job shop agile manufacturing environment, the working area of the machinery, cell , consist of multiple-purpose machines to deal with the product-mixed model. The processing of tasks in cell can be organized flexible job shop manner.

In this dissertation, we focus two canonical types of flexible and responsible scheduling systems considering product-mix and machine/tool allocation with variety of small batch jobs.

- **Type 1** General flexible job shop scheduling problem. it is one of the most popular scheduling models existing in practice which is among the hardest combinatorial optimization problems.

- **Type 2** Scheduling problems with sequence-dependent setup times. There has been a significant increase in interest in scheduling problems involving setup times. Since there are tremendous savings when setup times/costs are explicitly incorporated in scheduling decisions in various real world industrial/service environments.

As described in previous section, the scheduling problems in AMS can be thought as a sort of complex flexible job scheduling problem (FJSP), and the decision-making space can be considered as shown in Fig. 3.1:

- how to allocate the resources to each operation (*resource allocation*).
- how to perform the operations by an operation sequence (*operation sequence*).

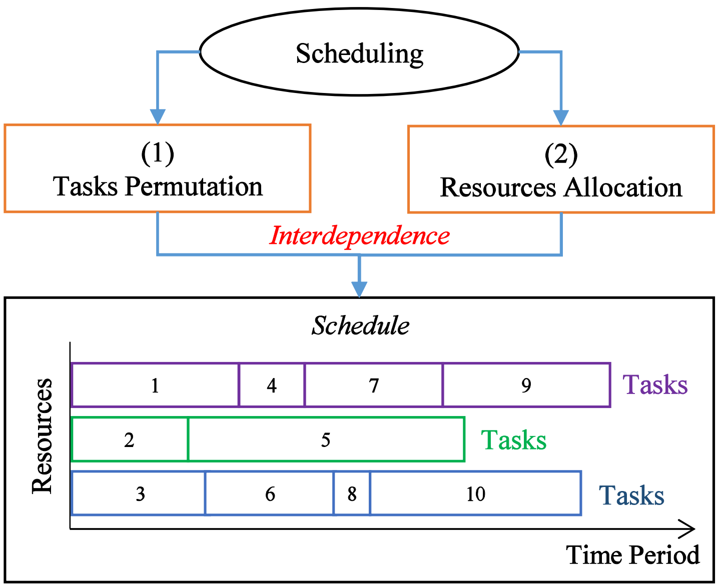


Fig. 3.1 Interdependence of resources allocation and operations sequence

The resource allocation determines which operation should be assigned to the suitable resource. Then, the decision of operation sequence will be made according to the previous phase's outcome. Consequently, the performance of scheduling is affected by machine allocation if the operations permutation was determined. In contrast, the effectiveness of resource allocation is depended on the operations permutation. Therefore, the interdependence of resources allocation and operation sequence make scheduling a formidable "combinatorial" problem. In general FJSP, researchers usually use two kinds of optimization approaches, (1) focus on the operations permutation, weakened the interdependence relationship between the resources allocation and operations permutation, such as single resource scheduling etc.; (2) focus on the resources allocation, diminish the effect in scheduling by operations permutation, general called resources allocation problems.

In the following section, we presents the interrelation of operations during the decision concerning resource allocation. In scheduling, the complete time of last job (makespan) and machine balancing is highly depending on a strategy of machine selection, as shown in Fig.3.2. This strategy is usually described as a machine allocation that each of operation of tasks is assigned to machine.

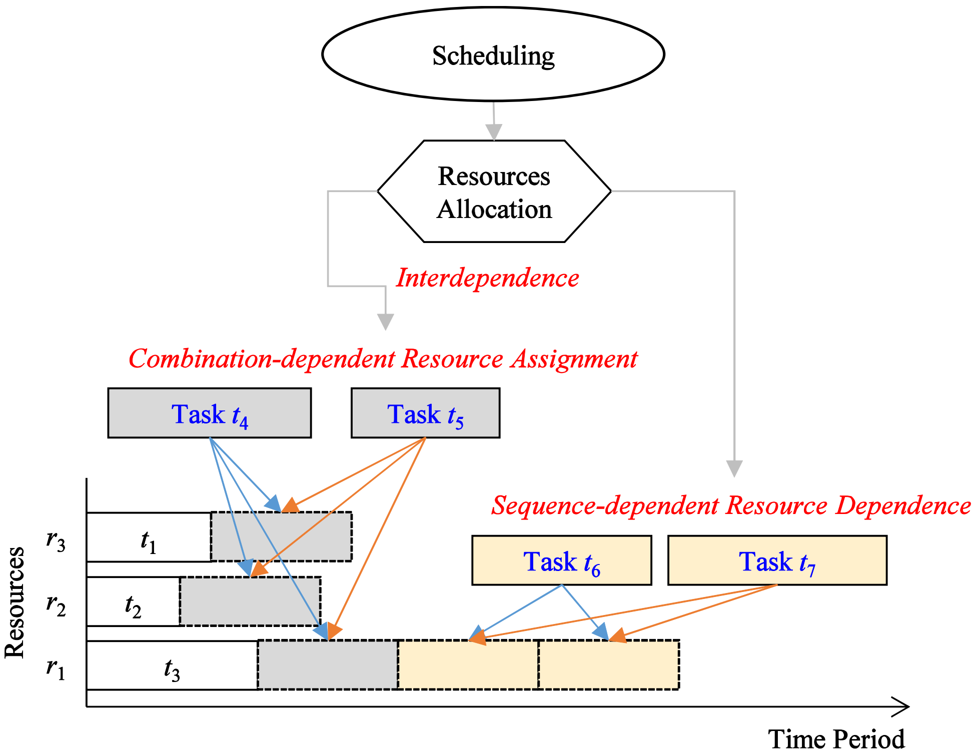


Fig. 3.2 Two kinds of interdependence of multiple resources

Therefore, in this dissertation, we will employ graphical probabilistic models (GPMs) which the interdependency between resource allocation and the character of sequence-dependent are taken into account. the primary task is to find:

How to build explicit probabilistic models for interdependent of variables?

The decision of machine assignment for operation can be made as follows

$$x_{ij} \sim p(x_{ij|N_{o_{ij}}}) \tag{3.1}$$

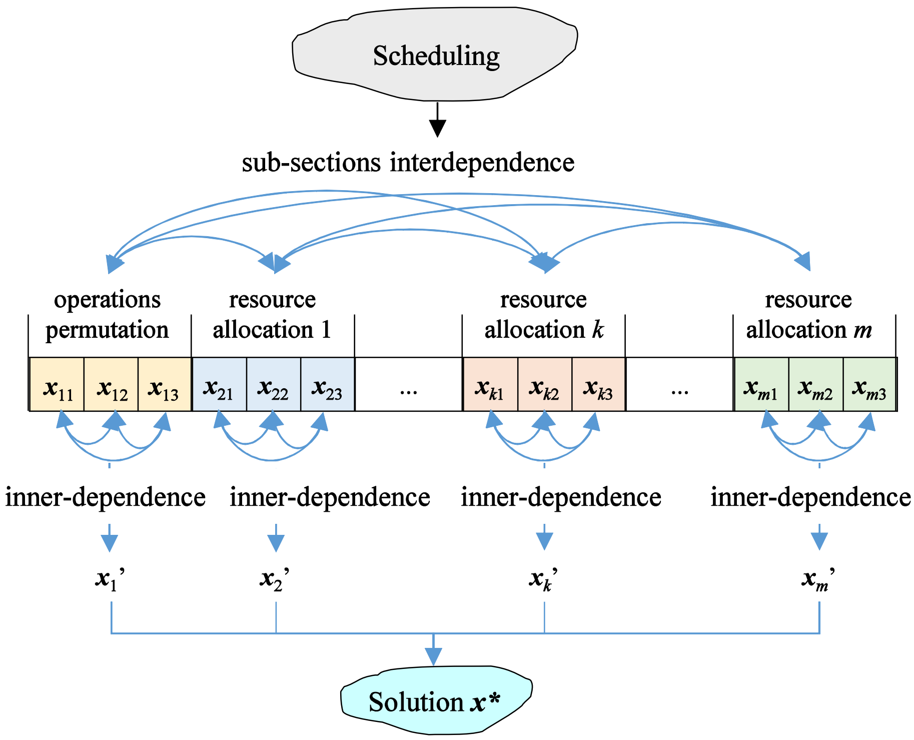


Fig. 3.3 Illustration of high-dimensional interdependence in scheduling problem

Where $N_{o_{ij}}$ denotes the neighborhood of operation o_{ij}.

On the other hand, building and learning an explicit probabilistic model for the problems should be able to pay a small price of the computation time complexity. Unfortunately, the most of scheduling problems, such as FJSP can be formulated as high-dimensional optimization problems. Moreover, learning the exact graphical probabilistic model is also NP-hard problem [79–81]. Therefore, in this dissertation, another task is to find :

How to learn an adequate probabilistic model within the considerable computation costs?

3.2 Our proposal

3.2.1 Framework of our proposal

For the flexible scheduling defined in *Type 1*, it can be divided into two sub-problems: operation sequencing and machine allocation. Herein, operation sequencing determines the processing order of operations and outputs sequence S for a schedule. Most of the literatures always prefer to determine the machine and related resources such as tools of operations with respect to constraints on jobs. Thereafter, the operation sequence S is constructed according to the previous machine decision and resources decision, it can be written as

$$D(X_{ij}) \rightarrow D(Y_{ij}) \rightarrow S \tag{3.2}$$

where $D(X_{ij}),D(Y_{ij})$ denotes the machine assignment, resource selection of operation o_{ij} respectively.

Unfortunately, for the flexible scheduling defined in *Type 2*, The overhead concerning of job moving and decisions on tools within each machine are crucially influenced by the immediately preceding decision in operation sequence, It is referred to sequence-dependent [82] and it increases the potential of changeover when the sequence-dependent constraint is not sufficiently considered ahead. Therefore, in this paper, we adopt the following decision approach :

$$S \rightarrow D(X_{ij}) \rightarrow D(Y_{ij})) \tag{3.3}$$

In this dissertation, Incorporating machine learning technique into EDA for combinatorial problems optimization has motivated our pursuit for developing a uniform framework named Hybridized probabilistic graphical models based EDA (h-PGMEDA) heuristic searching architecture, which integrates PGM model to predict the interdependence among decision variables by using machine learning techniques.

For improving the performance, the local search is provided by a suitable hybridization using problem-specific solvers including heuristics, approximate, and exact algorithms. Essentially, local search algorithms try to find high quality solutions by searching through the solution space. More precisely, these algorithms start with an initial solution, and then iteratively generate a neighboring solution in terms of small changes that may be applied to a solution. It is widely accepted that a local search procedure is efficient in improving the solutions generated by the EDA. The framework of h-PGMEDA is shown in Fig. 3.4.

As described in Fig. 3.4, PGMs provides a model to express the conditional dependence structure between variables. Two branches of graphical representations of distributions are commonly used, namely, Bayesian networks and Markov random field. Bayesian network via a Directed Acyclic Graph (DAG) to provide a set of variables and their conditional dependencies. It can represent a sequential competition in game theory. Markov random field (MR) an undirected graph with is a set of variables which have the Markov property. Therefore, our

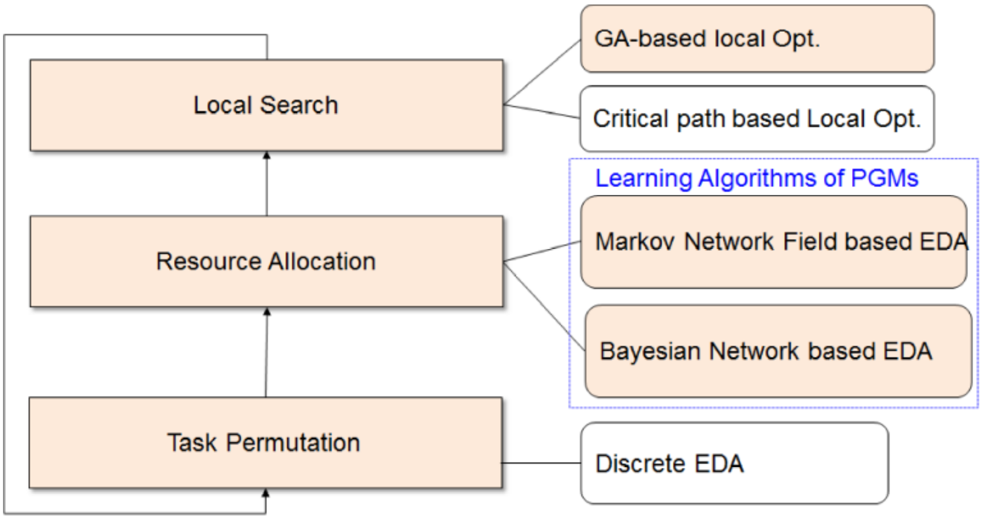


Fig. 3.4 The framework of the Hybridized PGM-based EDA

main idea is incorporate probabilistic graphical model-based learning techniques modeling multiple-dependent of the resource allocation in AMS.

h-PGMEDA starts by generating some solutions randomly which stored in the population. Several solutions with higher fitness chosen by a selection method from the population are always called the promising solutions, use them to learn the structure of PGM. Then, the conditional probabilities for each decision variable are estimated. Consequently, the new solution is generated according to the given sampling method. In order to achieve better performance, the generated candidate solution is improved the local optimal solution by problem-specific local search algorithm. Finally, new solutions are evaluated, and the solution with higher fitness are put into archive as the reference of promising data. The process above will go on until the termination criteria are reached. Fig.3.5 presents pseudo-code of Hybridized PGM-based EDA(h-PGMEDA).

3.2.2 Operation sequencing Algorithm

Essentially, operation sequence problem is tradition permutation problem which is one of classical optimization problem in flow-shop environment and job-shop environment. Since 2009, Jarboui proposed the discrete EDA for minimizing the total flow-time in permutation flow-shop scheduling problems [63]. It uses the univariate marginal distribution to estimate the marginal probability while taking into account the order of the job in the sequence. The

Hybridized PGM-based EDA (h-PGMEDA)

Define:
$X = (X_1, X_2, \ldots, X_n)$: vector of the decision variables;
$popSize$: number of the population solutions kept by h-PGMEDA;
gen: number of the generations (iterations);
$prRate$: rate of the promising solutions kept by h-PGMEDA;
$elitRate$: rate of elitism kept as the promising solutions;
end
Input: problem data, parameters
Output: s: the best solution

begin
  Initialization
    step 1: $t \leftarrow 0$;
    step 2: Initialize the population $pop(t)$ with the size $popSize$ solutions; evaluate $pop(t)$, and set the archive $A(t) = \{\}$;
  end
  while *not terminating condition* **do**
    step 3: Select the promising data set D from $pop(t)$ consisting of the best $popSize \times (1 - elitRate)$ solutions and $A(t)$;
    step 4: Update $A(t)$ with the best $popSize \times elitRate$ solutions from $pop(t)$ and $A(t)$;
    step 5: Estimate the structure of PGM represents the neighborhood N over the set D;
    step 6: Estimate the parameter of conditional probabilities $p(x_i|N_i)$ of PGM, for each variable X_i over the set D;
    step 7: Sample *candidates* based on $p(x_i|N_i)$ of PGM;
    step 8: Perform a problem-specific local search *DoLocalSearch*(*candinates*);
    step 9: Evaluate the candidates and update the best solution s;
    step 10: $t \leftarrow t + 1$;
  end
  Output: return the best solution s;
end

Fig. 3.5 pseudo-code of the proposed Hybridized PGM-based EDA (h-PGMEDA) main factor of the probability is estimated by the following equation:

$$p_{ik} = \frac{\mu_{jk}}{\sum_{l \in \Omega_k} (\mu_{jk})} \tag{3.4}$$

where μ_{ik} is the number of times of appearance of job i before or in the position k in the subset of the selected sequences, and Ω_k the set of jobs not already scheduled until position k. In our research work, we employ the same method solving operation sequence problem in order to compare with the previous approach fairly.

3.2.3 Learning Algorithm of PGMs

Markov random field-based Learning Algorithm

For the flexible job shop problem (Type 1), each job i is constitutive of a series of n_i operations $(o_{i1}, o_{i2}, ..., o_{in_i})$. FJSP allows an operation was performed on any particular machine picked up from a given set A_{ik}. The processing time on machine j of operation o_{ik} is p_{ijk}. The FJSP problem is choosing a machine, and the decision variable starting time s_{ik} is made at which each operation o_{ik} must be performed. Therefore, FJSP has more complex strategy than JSP (i.e., for each operation, the machine needed to be addressed.), the machine allocation (namely, in addition to the traditional scheduling decisions. The starting time, to make sure each operation machine rather than a different factor).

Based on the JSP assumptions, Assumptions of FJSP are described as follows:

A8. The precedence relation between operation work is predetermined, not violated.

A9. The processing time has already contains the set up time.

A10. Operation cannot be interrupted.

A11. Each operation can be seen as any machine is carried out. When an operation can't processing machines, machine processing time of the operation will be given a large positive value.

The symbols and notations stating mathematical model are described as the followings.
Notation:

Indices:

$$i \quad \text{job indexs}, i = 1, 2, \ldots, n \tag{3.5}$$

$$j \quad \text{machine indexs}, j = 1, 2, \ldots, m \tag{3.6}$$

$$k \quad \text{operation indexs}, k = 1, 2, \ldots, K_i \tag{3.7}$$

Parameters:

$$n \quad \text{job numbers} \tag{3.8}$$

$$m \quad \text{machines numbers} \tag{3.9}$$

$$J_i \quad \text{the } i\text{-th job} \tag{3.10}$$

$$K_i \quad \text{operation numbers of job } i(\text{ or } J_i) \tag{3.11}$$

$$o_{ik} \quad \text{the } k\text{-th operation of job } i(\text{ or } J_i) \tag{3.12}$$

$$M_j \quad \text{the } j\text{-th machine} \tag{3.13}$$

$$U_{ik} \quad \text{available machine set of the operation } o_{ik} \tag{3.14}$$

$$W_j \quad \text{workloads of machine } M_j \tag{3.15}$$

$$p_{ikj} \quad \text{processing time of operation } o_{ik}\text{c on machine} j(\text{ or } M_j) \tag{3.16}$$

$$U \quad \text{machine set with the size } m \tag{3.17}$$

Decision Variables

$$x_{ijk} = \{0, 1\} \quad \begin{cases} 1 & \text{if machine } j \text{ is choosed to process the operation } o_{ik}; \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

$$c_{ik} \quad \text{the completion time of the operation } o_{ik} \tag{3.19}$$

Formulation:

The FJSP mathematical model with minimizing the objective: makespan, is given as follows:

$$
\begin{aligned}
\min \quad & t_M = \max_i \max_k \{c_i k\} && (3.20)\\
\text{s.t.} \quad & c_{ik} - t_{ikj}x_{ikj} - c_{i(k-1)} \geq 0, k = 2, \ldots, K_i, \forall i, j && (3.21)\\
& \sum_{j=1}^{m} x_{ikj} = 1, \forall k, i && (3.22)\\
& x_{ikj} \in \{0, 1\}, \forall j, k, i && (3.23)\\
& c_{ik} \geq 0, \forall k, i && (3.24)
\end{aligned}
$$

Equation (3.21) states that the continuous operation started the precedent of the same operation after the completion of work, on behalf of the operator precedence constraints. In other words, the precedence constraint on the jobs can not be violated. Equation (3.22) states that must select a machine for each operation.

We consider FJSP with the n jobs and m machines show in Fig. 3.6. For each operation o_{ij}, there is a set of alternative machines on which the operation can be processed. From the machine assignment of view, the multiple-purpose machine can perform a set of operations which belongs to different job or not. Let O_m be the set of operations which are assignable to machine m. Considering the criterion makespan, the decision of machine selection of operation in O_m has implicitly affect the decision of the other operation in O_m. Extremely, all of the operations does be assigned to the same machine. However, in the EDA-based approaches given in section 2.1.2, the probabilistic model just learning the marginal distribution of the machine assignment, while the influence between operations in O_m is seldom taken into account. Let x_{ij} be decision variable concerning machine assignment of operation o_{ij}. The parameter of probabilistic model is estimated as follows:

$$x_{ij} \sim p(x_{ij}|N_{ij}) \tag{3.25}$$

It can represent a simultaneous competition in game theory. Therefore, based on the above description, PGMs are great approaches for solving the resource competition problems in scheduling. Therefore, this resource type is named combination dependent resources, and we can solve the resources competition problems by discover the relationship between the operations and resources, as shown in Fig.3.7. for the operations that is assignable to the same resources, the resource assignment per operation explicitly affect the remain that is not

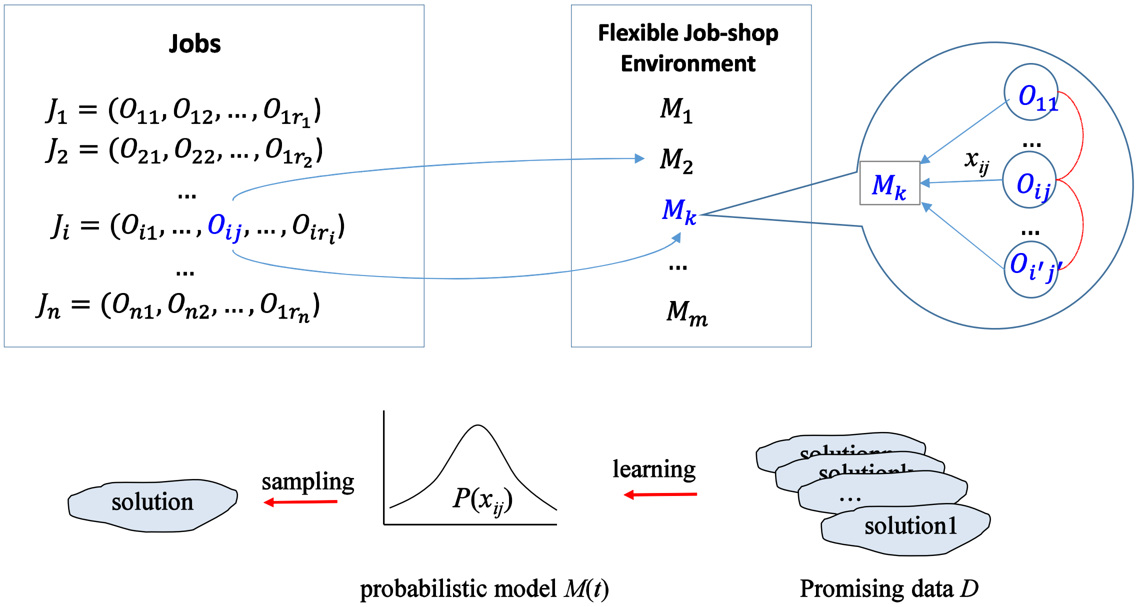


Fig. 3.6 Illustration of interdependence of machine allocation in FJSP (Type 1)

decided with respect to makespan criterion. Moreover, we have no knowledge concerning the priority of resource allocation.

We formulate machine allocation into a bi-partite graph model and solve it by using MR-based approaches.

Therefore, we propose effective Markov Random Field-based learning algorithm where MR models the interrelation of combination dependent Resources allocation. the framework is shown in Fig. 3.7.

Bayesian network -based Learning Algorithm

For the flexible job shop problem (Type 2), we consider the flexible job shop scheduling problem where the setup time of operation is not negligible. The main purpose of FJSP (Type 1) is to determine the process route of a job, while operations involved are allowed to any one of the multiple available machine candidates [83]. However, machine availability is the only constraint considered in the traditional FJSP. A more realistic scheduling model machine to other relevant resource constraints should be considered. In this study, we consider the FJSP having unlimited resource constraints on tools and tool approach directions (TADs) as found in some production systems.

The illustration of the FJSP (Type 2) problem in AMS is shown in Figure 3.8.

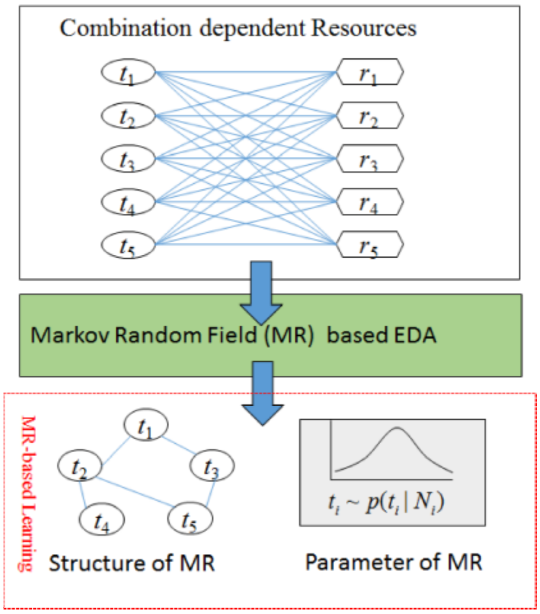


Fig. 3.7 Framework of Effective MR-based Learning Algorithm

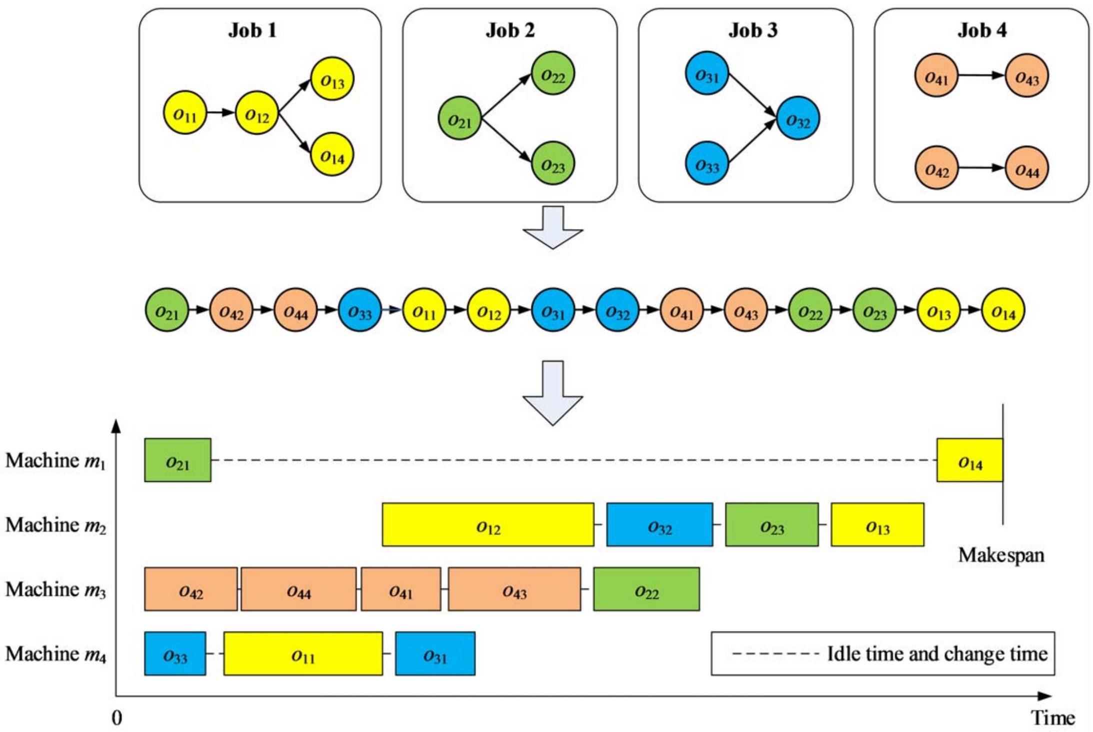


Fig. 3.8 Illustration of FJSP (Type 2) problem

Fig. 3.8 shows the process of four jobs processed on four machines with the tools and TADs. Each job is consisted of different numbers of operations (marked by colors respectively), each operation can be performed on a set of available machines with different processing time. The detail information of different operations of jobs is present in Table 3.1. From the first column, it presents the operation ID, successors, operation name, TAD candidates, machine candidates, tool candidates and machining time respectively.

The problem here is how, when, and to effectively allocate the operation sequence of work suitable manufacturing resources and to realize the objective of a given process to maintain the feasibility plan and schedule.

This problem can be defined as follows:

- Process planning: Resources such as machines, tools, and TADs are selected according to the geometry features and availability of machining resources, Fig. 3.9 shows an illustration of the decision process on resource selection of the job.
- Operation sequencing: Decide the executing sequences of all the operations required for the jobs in order to make the precedence relationships of all operations are not violated.
- Jobs scheduling: Determine how and when to assign the manufacturing resources to the jobs with regard to the resources' sequence-dependent constraints (shown in Fig. 3.10).

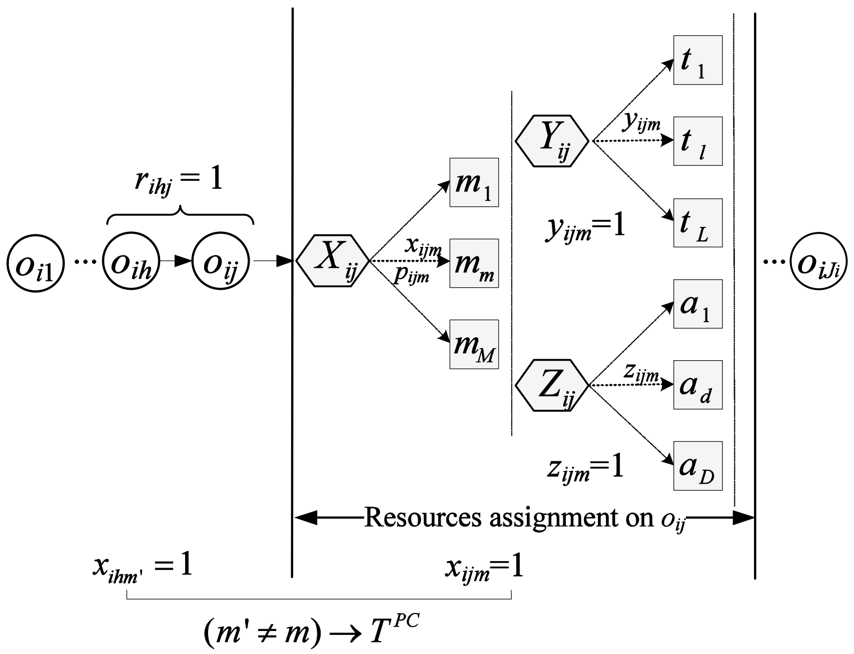


Fig. 3.9 Illustration of resource selection of job i

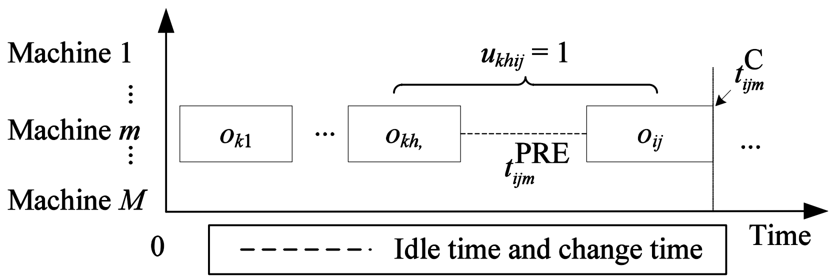


Fig. 3.10 Illustration of sequence-dependent constraints about resources

Table 3.1 Operation information of illustrative FJSP (Type 2) problem

Job-ID	**Op-ID**(o_{ij})	**Successor**	**Operations**	**Machine Candidates**(X_{ij})	**Tool Candidates**(Y_{ij})	**TAD Candidates**(Z_{ij})	**Machining time) (time unit)** (p_{ijm})
Job 1	o_{11}	o_{12}	Milling	m_2, m_4	t_6, t_7, t_8	+z	40, 30
	o_{12}	o_{13}, o_{14}	Milling	m_3, m_4	t_6, t_7, t_8	-z	40, 30
	o_{13}	–	Milling	m_2, m_4	t_6, t_7, t_8	-x	20, 15
	o_{14}	–	Drilling	m_2, m_4	t_2	+z, -z	12, 10
Job 2	o_{21}	o_{22}, o_{23}	Drilling	m_1, m_2, m_3, m_4	t_1	+z, -z	12, 10, 10, 7.5
	o_{22}	–	Milling	m_2, m_3, m_4	t_{12}	-x, +y, -y, -z	20, 20, 15
	o_{23}	–	Milling	m_2, m_3, m_4	t_5, t_6, t_{11}	+y	18, 18, 13.5
Job 3	o_{31}	o_{32}	Milling	m_2, m_3, m_4	t_6, t_7, t_8	+z	20, 20, 15
	o_{32}	–	Milling	m_2, m_3, m_4	t_6, t_7, t_8	-z	20, 20, 15
	o_{33}	o_{32}	Milling	m_2, m_3, m_4	t_6, t_7, t_8	+x, -x, +y, -z	15, 15, 11.25
Job 4	o_{41}	o_{43}	Milling	m_2, m_3	t_6, t_9	-y	12,15
	o_{42}	o_{44}	Milling	m_2, m_3	t_9, t_{10}	-y	21,18
	o_{43}	–	Milling	m_2, m_3	t_3	-z	18,25
	o_{44}	–	Milling	m_2, m_3	t_1, t_3	+x, -x	27,25

For each job, the operation sequences have precedence constraints, and the process can not be violated in before the job completed.For example, a feasible solution with 14 operations,the operation sequence is listed as (o_{21}-o_{42}-o_{44}-o_{33}-o_{11}-o_{12}-o_{31}-o_{32}-o_{41}-o_{43}-o_{22}-o_{23}-o_{13}-o_{14}). Determine the corresponding resources for operations in order to form the completed solution. FJSP (Type 2) is subjected to the followings constraints:

A1. Each machine can only handle one operation at a time;

A2. Each operation is completed before another operation is loaded;

A3. The sequence of the operations of each job complies with the manufacturing constraints;

A4. All jobs, machines and tools are available at time zero;

A5. Each operation is performed on a single machine, and each machine can only execute an operation at a time;

A6. The set-up time is identical and independent for specific operations. The time for a machine changeover or a tool changeover follows the same value;

A7. Machines are continuously available for production.

Notation

There exists a lot of objectives of FJSP (Type 2) optimization problems which contains makespan, total cost, workload, processing time and so on. Minimizing makespan is a frequently-used and important objective among them. The core idea of minimizing makespan is that limit the complete time of all jobs as short as possible.The notations used to express the mathematical model is listed in the following.

Indices:

i,k	indices of jobs, $(i,k = 1,2,\dots,I)$
j,h	indices of operations for job i,$(j,h = 1,2,\dots,J_i)$.
m	index of machines, $(m = 1,2,\dots,M)$.
l	index of tools, $(t = 1,2,\dots,L)$.
d	index of TADs, $(d = 1,2,\dots,D)$.

Parameters:

I	number of jobs.
J_i	number of operations for job$i$.
M	number of machines.
L	number of tools.
D	number of TADs.
o_{ij}	j-th operation of job$i$.
m_m	$m-$th machine.
t_l	$l-$th tool.
a_d	$d-$th TAD.
r_{ijh}	precedence constraints.
	$r_{ijh} = 1,$ if$o_{ij}$is predecessor ofo_{ih}; $0,$ otherwise.
p_{ijm}	processing time of operation$o_{ij}$by machine$m$.
t^{PC}	machine changeover time, it is used to express the lead time when two adjacent operations of the same job is transferred between different machines.
t^{TC}	tool changeover time, it considers the tool changeover time of two adjacent operations carried out on the same machine.
t^{DC}	TAD changeover time, it considers when the TADs of two adjacent operations on the same machine is different.
t_{ijm}^{PRE}	preparation time of operation$o_{ij}$by machine$m$. The preparation time for an operation consists of machine change time, tool change time and set-up time for the operation.
t_{ijm}^{PRE}	$= t^{\mathrm{PC}} + t^{\mathrm{TC}} + t^{\mathrm{DC}}$
X_{ij}	set of alternative machines that can processo_{ij}.
Y_{ij}	set of alternative tools that is used to processo_{ij}.
Z_{ij}	set of alternative TADs that is used to process by machine$m$.
t_{ijm}^{C}	completion time of operation$o_{ij}$by machine$m$.

Decision Variables:

$$x_{ijm} = \begin{cases} 1, & \text{if } o_{ij} \text{ is performed on machine } m \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ijl} = \begin{cases} 1, & \text{if } o_{ij} \text{ is performed on tool } l \\ 0, & \text{otherwise} \end{cases}$$

$$z_{ijd} = \begin{cases} 1, & \text{if } o_{ij} \text{ is performed on direction } d \\ 0, & \text{otherwise} \end{cases}$$

$$u_{ijkh} = \begin{cases} 1, & \text{if } o_{ij} \text{ is performed directly before } o_{kh} \\ 0, & \text{otherwise} \end{cases}$$

Formulation

The mathematical model for minimization of makespan can be stated as follows:

$$\min \quad t_M = \max_{i,j,m} t^{\mathrm{C}}_{ijm} \tag{3.26}$$

$$\text{s.t.} \quad \left(t^{\mathrm{C}}_{khm} - p_{khm} - t^{\mathrm{PRE}}_{khm} - t^{\mathrm{C}}_{ijm}\right) x^{M}_{ijm} x^{M}_{khm} u_{ijkh} \geq 0, \qquad \forall (i,j),(k,h),\forall m \tag{3.27}$$

$$r_{ijh} u_{ihij} = 0, \forall (i,j), h \tag{3.28}$$

$$u_{ijij} = 0, \forall (i,j) \tag{3.29}$$

$$\sum_{m=1}^{M} x_{ijm} = 1, \forall (i,j) \tag{3.30}$$

$$\sum_{l=1}^{L} y_{ijl} = 1, \forall (i,j) \tag{3.31}$$

$$\sum_{d=1}^{D} z_{ijd} = 1, \forall (i,j) \tag{3.32}$$

$$x_{ijm} = 0, \forall (i,j) \notin X_{ij}, \forall m \tag{3.33}$$

$$y_{ijl} = 0, \forall (i,j) \notin Y_{ij}, \forall l \tag{3.34}$$

$$z_{ijd} = 0, \forall (i,j) \notin Z_{ij}, \forall d \tag{3.35}$$

$$u_{ijkh} \in \{0,1\}, \forall (i,j),(k,h) \tag{3.36}$$

$$x_{ijm}, y_{ijl}, z_{ijd} \in \{0,1\}, \forall m, d, l, (i,j) \tag{3.37}$$

$$t^{\mathrm{C}}_{ijm} \geq 0, \forall m, (i,j) \tag{3.38}$$

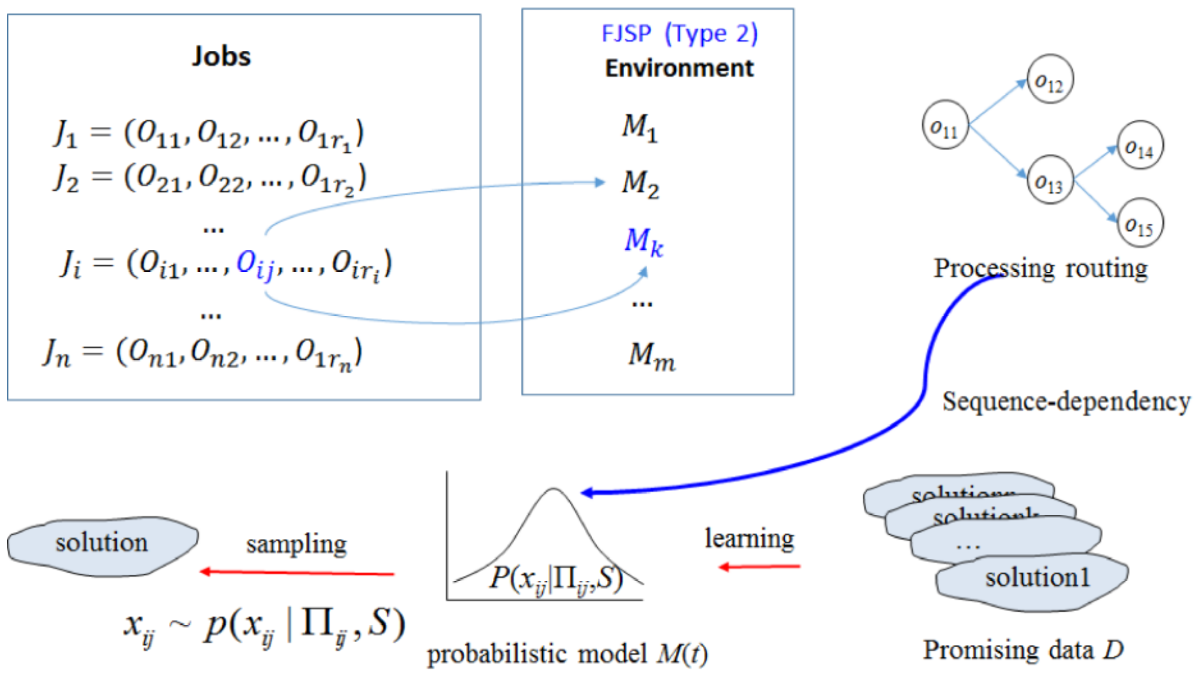


Fig. 3.11 Illustration of interdependence of machine allocation in FJSP (Type 2)

Equation 3.26 describes the objective for the minimization of makespan t_M. Equation 3.27 imposes that any machine cannot be selected for one operation until the predecessor is completed. Equation 3.28 ensures that the precedence constraints are not violated. Equation 3.29 ensures the feasible operation sequence. Equations 3.30, 3.31, 3.32, 3.33, 3.34, and 3.35 ensure the feasible resource selection. Equations 3.36, 3.37 and 3.38 impose non-negative conditions.

From the illustration of interdependence of machine allocation in FJSP (Type 2), the resource allocation not only considers the machine allocation of decided operations, but refers to operation sequence. Therefore, we formulate sequence-dependent machine allocation and resource selection into a sorting problem and solve it by using BS-based learning approaches where . The framework is shown in Fig. 3.12.

3.2.4 Local Search

For h-PGMEDA, the performance highly depends on the accuracy of PGM. In h-PGMEDA, EDA pays more attention to global exploration, in order to avoid losing the diversity, we taken into account local search strategy to enhance the exploitation capability. We consider

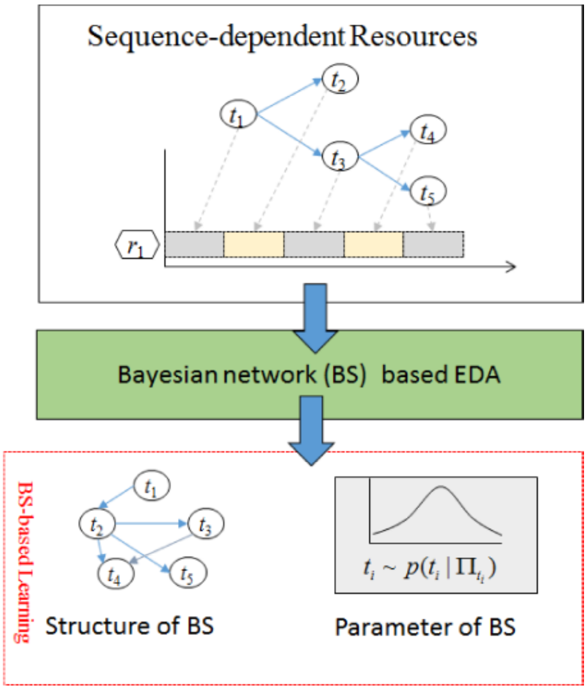


Fig. 3.12 Framework of Effective BS-based Learning Algorithm

critical path-based local optimization and GA-based local optimization respectively with respect to the problem-specific features.

Table 3.2 Vital components of Learning algorithms of PGMs

	Markov Random Field EDA (MREDA)	Bayesian network EDA (BSEDA)
Structure Learning	Neighborhoods G_{MR}	Parenthoods G_{BS}
Parameters Estimation	Gibbs Measure: ψ_{cpt}	Maximum likelihood θ_{cpt}
Alternative Solution Generation	Gibbs Sampling	Probabilistic Logic Sampling

As show in Fig. 3.4, The vital components of Learning algorithms of PGMs are :

- Learning the structure of PGMs
- Estimating the parameters of PGMS
- Sampling alternative solution according to PGMS

The vital components of h-PGMEDA are presented in the following subparagraph in detail with respect to Bayesian network (BS) and Markov random Field (MR). Firstly, the structure learning of PGM is described. Then, the parameter estimation corresponding to edges is presented. Finally, sampling procedure is introduced in detailed.

Chapter 4

Markov random field-based EDA (MREDA) and its scheduling application

scheduling problems can be generally and well considered as "resource allocation problem with the passage of time and a series of tasks" [84]. Survey of scheduling research is referred as widely diverse scheduling problems [85, 86]. Particularly, the job shop scheduling problem (JSP) is introduced to deal with the sequencing decision of manufacturing operations in which job routes are determinate. JSP is one of the well-known hardest combinatorial optimization problems in which the operation of the machine sequence determination. It is made up of several assumptions as the following: [58]:

A1. The process of each machine is only one job.

A2. Every job is processing at the same time on only one machine.

A3. Not able to access the same machine work twice for each job.

A4. Processing time for each operation have been given in advance.

A5. There is no preference constraint in the operation of different tasks.

A6. Operations cannot be interrupted.

A7. Release time and deadline are not specified.

This problem has been substantiated that it is one of the most notoriously intractable NP-hard problem. The traditional JSP considers n-jobs and m-machines, each work consists of a set of operating the machine and operation sequence is predetermined, and considering the each operation on the machine processing and processing of the fixed time is processed on the given machine and deal with the fixed time [87, 87, 56].

Flexible job-shop scheduling problems (FJSP) are always thought as an extended from the traditional JSP, where a machine may have the ability to perform more than one type of operation [88]. In other words, for any given, at least, one machine is capable of handling the operation. In FJSP problem, there are two types of flexibility to give a description of the performance FJSP problem. [89]:

- *Partial flexibility*: Only some action on the part of the available machine can be realized.

- *Total flexibility*: All actions can be processed on all the available machines.

Over the last three decades, there is a growing number of efforts focusing on solving the FJSP and appreciable amount of papers have been published, unfortunately, there is no effective solution algorithm to solve the optimality in polynomial time by now.

Brucker and Schlie research work firstly address the FJSP [90]. In that paper, they solved two-jobs FJSP, if to solve a problem with more complex, in can use two kinds of methods: one type is hierarchical methods and the other is comprehensive methods which are well studied to solve the FJSP. The first is a hierarchical or a decomposition approach which solves the machine assignment problem first; and based on its solution, a JSP is formulated and solved. The second is a concurrent one which looks for a solution to both problems simultaneously. Early studies of the FJSP developed both hierarchical and concurrent approaches that are based on mixed integer linear programming (MILP) models and dispatching rules. In the last three decades, many researchers found the FJSP fertile for emerging meta-heuristic approaches since exact algorithms are computationally prohibitive and the capabilities of traditional heuristic approaches are limited.

Recently, Wang *et al* presented bi-population based estimation of distribution algorithm (BEDA) for solving the FJSP with the criterion to minimize the makespan[100]. In BEDA, the univariate marginal distribution model (UMDM) is used to estimate the probability that operation o_{ij} is processed on a machine. Unfortunately, BEDA does not consider the interaction effects within assigning machines of the operations that a machine can perform. The interactions have an influence on potentially minimizing the objective makespan.

In this chapter, we proposed Markov random field-based EDA (MREDA). In MREDA, Markov network is used to model conditional dependency of multi-dimensional variables in EDA. MREDA takes care of exploration that tries to identify the most promising search space regions, and to model conditional dependence of the multi-dimensional variables. The conditional probabilities defined by the local Markov property are estimated, and the new candidate solutions are sampled according to the given sampling method. For a candidate solution, problem specific based local search algorithm is used to improve each candidate solution to reach a local optimum.

4.1 Efficient Markov Random Field Learning Algorithm

4.1.1 Structure learning

Currently, a majority of Markov Random Field based EDAs such as DEUM proposed by shakya [91] are to factorize the joint probability distribution represented in the equation 2.8. According to graphical theory, the cliques of the undirected graph are recognized. Then, the new solution is generated with respect to the estimated structure. Unfortunately, they suffer from the curse of dimensionality, even for small cliques. Moreover, factorizing methods lead to a complex approximation involving cliques, and the performance highly depends on the definitions of potential functions. In our proposal, constructing the neighborhoods of a variable is utilized as an alternative to factorizing the joint probability distribution in the decision space according to the local Markov property.

To construct a network model by the domain experts knowledge for a certain area is a time-costing task. Recently, the learning of structures of MNs from data attracts lots of attentions among researchers. So, there be proposed many structure-learning methods containing the methods based on conditional independence tests[92]. The method uses an independence test based on the mutual information(MI) to estimate the structure to avoid the complex process to construct a complex network structure. The value of MI can by calculated as follows, where X and Y represent two random variables.

$$\mathrm{MI}(X;Y)=\sum_{\mathrm{y}\in\mathrm{Y}}\sum_{\mathrm{x}\in\mathrm{X}}\mathrm{p}(x,y|D)\log\left(\frac{p(x,y|D)}{p(x|D)p(y|D)}\right) \tag{4.1}$$

where the $p(x|D)$ is the marginal probability of $X=x$ and $p(y|D)$ is the marginal probability $Y=y$, $p(x,y|D)$ is the joint probability of $X=x$ and $Y=y$, the two kinds of probability are computed from D, and the sum is all possible combinations of random variables X and Y. In other words, the MI value can be obtained by the joint probability and the marginal probability of X and Y, and they have the detail value x and y respectively. We can use the value of EI among pairs of variables to infer the dependence between two random variables.

Step5: Estimating structure of Markov Random Field

Input:
X: $X = (X_1, X_2, \ldots, X_n)$, vector of the decision variables.
D: promising solutions determined by **Step 3**.
Output:
N: define a neighborhood system on X, where each N_i is the set of variables neighboring to the variable X_i.

begin
 Step 4.1: Generate a matrix of mutual information according to the decision variables X, and compute the MI between each node pair in the solution over the promising data D (using equation 3.4).
 Step 4.2: Create an edge between the variables, if the MI between them is higher than the given threshold calculated by equation 3.5.
 Step 4.3: If the number of neighbors to a variable reaches the maximum number of k-neighbors, the k-neighbors with the highest MI are kept.
end

Fig. 4.1 Pseudo-code for estimating structure of Markov Random Field

The supporting theory of using this method is that when two nodes are connected, there exists lots of information of one node on the state of the connected node. That is, there exist more dependent between the connected nodes than other any nodes in the network[93].

Fig. 4.1 presents the implementation of structure learning algorithm for Markov network.

The mutual information of each two variables is used to build a matrix stored the mutual information. For the decision variables $X = (X_1, X_2, \cdots, X_n)$, MI has a $n \times n$ matrix, and use the equation 4.1 to estimate the values of elements in matrix. When we choose the neighbors for one variable, if one pair has the mutual information with higher value than a threshold with a certain value given in advance, then, to link one edge between this pair of variables. The threshold value is calculated as follows in order to be the symmetry of MI :

$$\begin{aligned} \text{thresholdValue} &= \alpha \times avg(\text{MI}) \\ &= \alpha \times \frac{2 \times \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \text{MI}(X_i; X_j)}{n \times (n-1)}; \\ X &= (X_1, \ldots, X_n) \end{aligned} \tag{4.2}$$

in which $avg(\text{MI})$ is the mean value of Mutual information matrix, and the coefficient parameter α can decide the edges density in MRs. The bigger α will lead to fewer edges in estimated structure and the smaller α will lead to more edges in estimated structure. So,

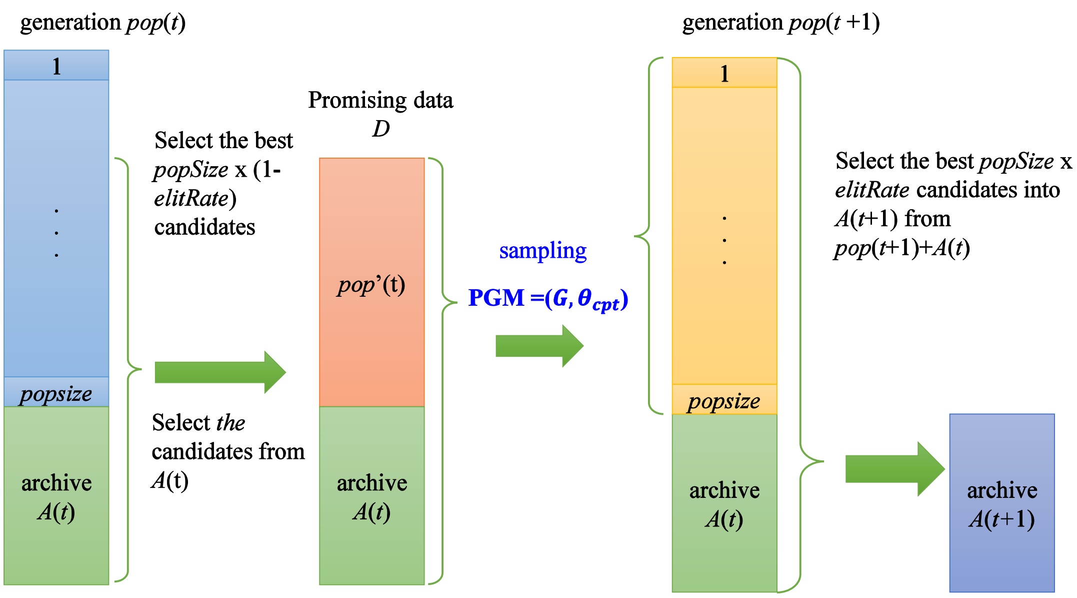


Fig. 4.2 Illustration of elitist archive-based structure learning

the size of sampling errors and the promising solutions will affect the accuracy of the model directly.In our experiments, α is first set to be 1.5.

The parameter k determines the network complexity. It determines the max number of neighbors, if k has a higher value, it means there needs more numbers of neighbors to be represented the conditional probability , so the number will directly affect the accuracy and the complexity of the estimated structure.

From the optimization of view, the size of promising data is always sparse (comparing to exact structure learning). Moreover, some of the noising data causes the over-fitting and bias of structure learning. Inspired by multi-objective optimizations, in our proposed, we involve the elitist archive mechanism which keeps some of reference solutions to improve the diversity of sampling and the accuracy of structure. Fig. 4.2 shows the process of structure learning.

4.1.2 Parameter estimation

Since Markov Random Field has the Markov property given in Section 2.2.2, the Gibbs distribution is an appropriate choice for the joint probability distribution following Hammersley–Clifford theorem . The conditional probabilities is calculated as Gibbs probability:

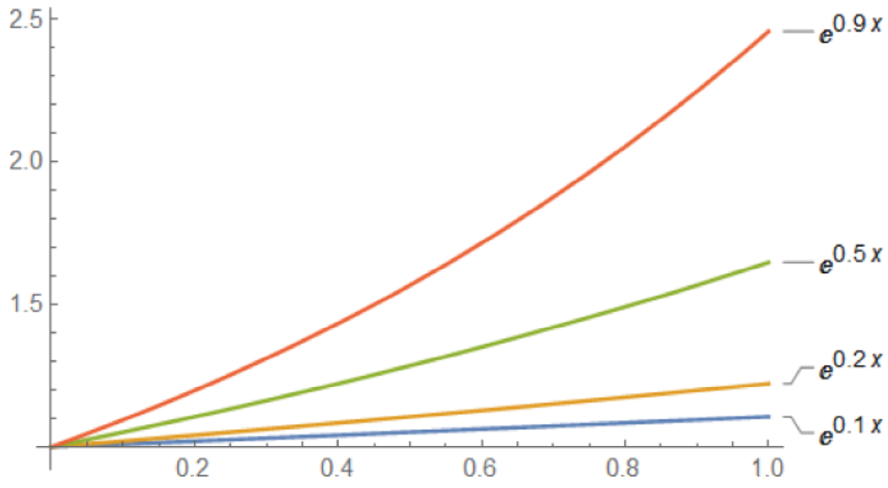


Fig. 4.3 Illustration of convergence speed under different cooling rate parameter

$$p(x_i|N_i) = \frac{e^{p(x_i,Ni)/T}}{\sum_{x_i^{\prime} \in X_i} e^{p(x_i^{\prime},Ni)/T}} \tag{4.3}$$

Where T is the temperature coefficient of gipps control convergence probability distribution.In this study, the linear table is used to represent a function of temperature $T \sim 1/gen \cdot \beta$, in which β represents the cooling rate parameter in MREDA, and gen represents the minimum number of generations. The Fig. 4.3 shows the convergence speed under different cooling rate parameter.

As depicted in Fig. 3.5, in MREDA, the size of the promising solutions greatly decides the learning pressure of a network($eliRate$) picked up from population. After getting the network structure, MREDA generate the new individuals according to the network for new solution searching space. The number of sampling from the network and the number of the individuals from the current population determine the convergence rate of MREDA. When the number of sampled individuals increase to a large value, diversity of the search space exploration will be lost. So, this leads to a bias to construct Bayesian network parameters often change or increase the likelihood of premature convergence. For adjusting the sampling number of new solutions, the probability of sampling on decision variable is incorporated. In other words, when MREDA begins the next evolutionary iterator, the parameters of Bayesian network will be reconstructed. It will balance the new parameters and the parameters got before, this is by the coefficient of parameters α. Theoretically, the probability of general random variable X in evolutionary iterator $t+1$ can be written as

$$MI(X_i;Y_j)_{t+1} = (1-\alpha)MI(X_i;Y_j)_t + \alpha\Delta(X_i;Y_j)_{t+1} \tag{4.4}$$

where $\Delta(X)_{t+1}$ represents the parameter of X, and the current promising solution will determine the value of the parameter. Particularly, when $\alpha = 1$, Bayesian network parameters will be completely rebuilt by the promising solution.

4.1.3 Sampling

When the MR is structured, we will use the structured network to generate new candidate solutions [94]. It is different from Bayesian network that, most Markov network does not meet the ancestor of most variable sampling probability logic needed for the order. So we choose an extended Gibbs sampling method which is one type of Markov Chain Monte Carlo (MCMC) method.[95].

The pseudo-code is described in Fig. 4.4, where the notation $p(x_i, N_i)$ represents the joint probability of a variable $X_i = x_i$ and the given value of neighbors N_i. We set β to 0.5, which aims to consider the balance between the exploitation and the exploration.

For instance, the structure shown in Fig. 4.5 is estimated, and the domain of variable is $0,1$. Following the procedure shown in Fig. 4.4. We illustrate how to sample new alternative solution. Firstly, a candidate solution is generated randomly, and let the solution be $x^{(1)} = (0,1,0,1,1)$. At step2, a permutation is generated l which determine the priority of decision-making, and let the priority be $l = (2,3,1,4,5)$. Then, for each variable, the decision variable sampling is executed. For the first variable x_2 given by the priority l, the value of x_2 is conditioned on $p(x_2, |N_2) = p(x_2|x_1, x_4)$. The second picked variable is x_3, and its value is conditioned on $p(x_3, |N_3) = p(x_3|x_1, x_4)$. The iteration is terminated utile the last variable x_5 is decided.

4.2 Design for FJSP (Type 1)

The following subparagraph presents the vital components of the proposal shown in Fig.3.5 in detail. Firstly, Markov Random Field is employed to encode the machine assignment of operations is described. Then, sampling new alternative solution based on the Markov network is presented. Finally, brief introduction of local search algorithm of Markov Random Field is summarized.

Step7: Gibbs Sampling based on Markov Random Field

Input:
$X = (X_1, X_2, \ldots, X_n)$: vector of the decision variables;
$popSize$: number of the population solutions;
D: promising solutions determined;
G: estimated structure of Markov Random Field;
N: a neighborhood system on X, where each N_i is the set of variables neighboring to the variable X_i;
Output:
$candidates$: set of new solutions created by Gibbs sampler;

begin
  for $j \leftarrow 1$ *to* $popSize$ **do**
    step 7.1: Generate a candidate solution $x^{(j)} = (x_1, x_2, \ldots, x_n)$ randomly according to the decision variables X;
    step 7.2: Generate a permutation l of a set $\{1, 2, ..., n\}$, let l_i be the value of i-th element in l;
    for $i \leftarrow 1$ *to* n **do**
      step 7.3: Choose a variable X_i from the solution $x^{(j)}$;
      step 7.4: Using the selected set of solutions D, estimate the conditional $p(x_i|N_i)$ for each value x_i of the variable X_i as Gibbs probability according to the following equation:

$$p(x_i|N_i) = \frac{e^{p(x_i,Ni)/T}}{\sum_{x_i' \in X_i} e^{p(x_i',Ni)/T}}$$

      step 7.5: Sample new x_i according to the conditional probability distribution $P(x_i|N_i)$;
    end
  end
end

Fig. 4.4 pseudo-code of Gibbs Sampling based on Markov Random Field

4.2.1 Markov Random Fields-based Encoding

There are two phases of the decision of FJSP. Firstly, to determine the operation sequences, in the process of which, use univariate marginal distribution to estimate the marginal probability, which means job *i* has the probability to be processed in the will be *k*-th one in the operation sequence. The method mentioned above is proposed in 2012, and is widely used in recent [96].

Secondly, to determine the machine allocation for each operation of different jobs, the decision set of *o* decision is likely to depend on each other in accordance with the relevant

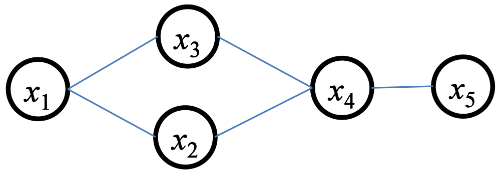


Fig. 4.5 Example of the estimated MR

business between the resources sharing. It is used to estimate the relationship of the machine allocation using the same machine. The node X_i is encoded by machine assignment decision variable Y_{ik} which is the k-operation o_{ik} of job i, as shown in MR (shown in Fig. 2.15). We use the Gibbs probability to estimate he conditional probability $p(y_{ik}|N_{ik})$ for each decision variable Y_{ik}:

$$p\left(y_{ik}|N_{ik}\right) = \frac{e^{p(y_{ik},N_{ik})/T}}{\sum\limits_{y_{i'k} \in A_{ik}} e^{p(y_{i'k'},N_{ik})/T}} \tag{4.5}$$

where y_{ik} is value of Y_{ik}, give the domain of Y_{ik} of A_{ik}, and N_{ik} on behalf of the neighbor node coding machine allocation decision variables Y_{ik}.

Giffler-Thompson heuristic algorithm[97] proposed decode method is used to produce a feasible solution when the operation sequences and the machine assignments has been confirmed according to the algorithms mentioned in the following. The feasible solution offers some time information containing the start time, the end time, the processing time and makespan. The pseudo code of decoding procedure is present in Algorithm 4.6.

4.2.2 LocalSearch

Local search in MREDA shown in Fig. 3.5 is used to improve the convergence speed and the quality of the solution. MREDA added a local optimizer and used for each individual before added to the original population.

a) Disjunctive Graph

A disjunctive graph $\mathscr{G} = (\mathscr{N}, \mathscr{A}, \mathscr{E})$ can be used to represet the feasible solutions of FJSP, in which $\mathscr{G}$ means node set, $\mathscr{A}$ means ordinary arc set , and $\mathscr{E}$ means disjunctive arc set. The nodes of $\mathscr{G}$ represent the corresponding operations, the real arcs $\mathscr{A}$ represent the relationships of precedence, and the dashed arc $\mathscr{E}$ represent the direct realization on the same machine

Giffler-Thompson algorithm based decoder

Input:
seq: operation sequence generated in terms of the algorithm
V : set of machine assignments, $V(o_{ij})$ denotes machine assignment of j-operation of job i.
Output:
S: the feasible schedule which consists of set of a pair of t^S_{ij}, t^C_{ij} for each operation o_{ij}, where t^M_m : the earliest available time of a machine item M_m; t^S_{ij} : start time of operation o_{ij}; t^C_{ij} : completion time of operation o_{ij}.
$C_{\max}$: makespan of the feasible schedule S.

begin
  Step 1: $t^M_m \leftarrow 0, \forall m$.
  Step 2: $t^S_{ij} \leftarrow 0, \forall o_{ij}$.
  Step 3: $t^C_{ij} \leftarrow 0, \forall o_{ij}$.
  for $i := 1$ *to* $|seq|$ **do**
    Step 4: $o \leftarrow v(x)$.
    Step 5: $m^* \leftarrow v(o)$.
    Step 6: $t^S_{ij} \leftarrow t^M_{m^*}$.
    Step 7: $t^C_{ij} \leftarrow t^S_{ij} + P_{ijm^*}$.
    Step 8: $t^M_{m^*} \leftarrow t^C_{ij}$.
    Step 9: $C_{\max} \leftarrow \max(C_{\max}, t^C_{ij})$.
  end
end

Fig. 4.6 pseudo-code of Giffler-Thompson algorithm based decoder

execution sequence of operations. In order to shown cleary, we choose a feasible solution with 3 jobs as an example, the graph and the Gantt chart are shown in Fig. 4.7 and 4.8.

In Fig. 4.8, *S* is the starting node, *T* is the terminating node, the pair of nodes of each nodes means the processing time and the selected machine for the linked job. For example, at the beginning time of machine 1, it processes o_{21} (the first operation of job 2) and after completing it, it begin to process o_{11} (the first operation of job 1).

The job predecessor $PJ(r)$ and the machine predecessor $PM(r)$ are moved respectively before *r* and the machine allocation of *r* in the same solution. For two node *r* and *v* ,they are both their job predecessor and machine successor, as the same as machine predecessor. In Fig. 4.8, if *r* sets o_{12}, the $PJ(r)$ is o_{11}, and the $SM(r)$ is o_{13}.

The task of local search task is to determine and divide one by one of the critical path for getting a new schedule with shorter makespan. If the operation *r* is critical, it need at least one of existing $PJ(r)$ or $PM(r)$ to be critical. In this paper, if both $PJ(r)$ and $PM(r)$ exist,

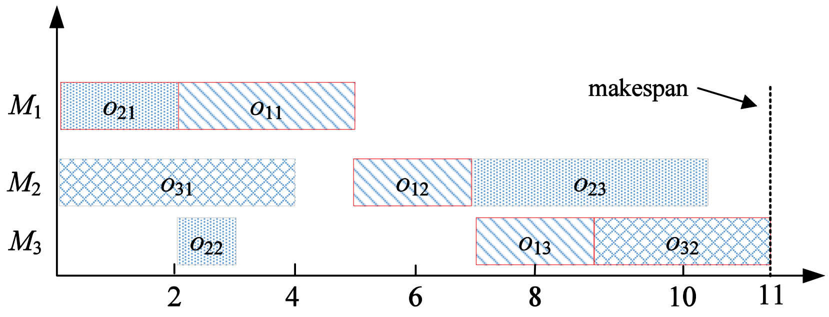


Fig. 4.7 Relative Gantt chart of a feasible solution

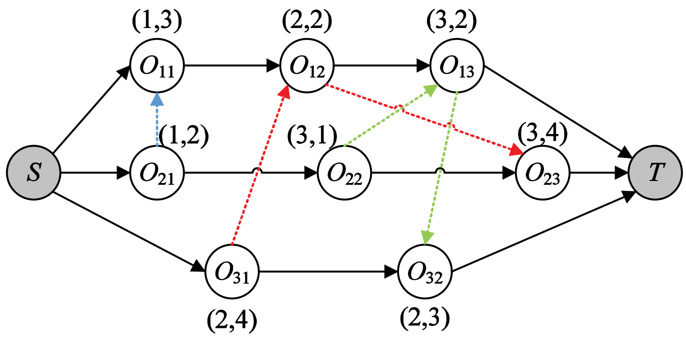


Fig. 4.8 Illustration disjunctive graph of a feasible solution

we give a rule that choose the job predecessor. Only a random selection of the critical path P of the disjunctive graph $\mathscr{G}$ can be considered to reduce the computing workload. As shown in Fig. 4.8, use the red line to mark out the operations which forms a critical path in Fig. 4.7, in which it contains the operation sequences $\{\ o_{21}, o_{11}, o_{12}, o_{13}, o_{32}\ \}$.

In this chapter, we proposed the variable neighborhood search (VNS) based on Markov network which aims to improve the quality of the optimal solutions, inspired by a thought which considers the change of neighborhood systematically within a local search with certain possibility [98].

b) Moving One Operation

Firstly, consider the method of moving the operations in the critical path. The method is to change the current position of the operation r to another position in disjunctive graph $\mathscr{G}$. The method of deleting r is removing the arcs between r, connecting $PM(r)$ and $SM(r)$ using a dotted line, meanwhile make r have the processing time ’0’. The disjunctive graph after removing operations for the $\mathscr{G}$, use $\mathscr{G}^-$ represent it. Use $C_M(\mathscr{G})$ represent the makespan of

$\mathscr{G}$. Due to the difference between $\mathscr{G}^-$ and $\mathscr{G}$ is deleting one operation from $\mathscr{G}$,obviously, the makespan of $\mathscr{G}^-$ will not be larger than $C_M(\mathscr{G})$. The description is shown in detail in Algorithm 4.9.

Critical path-based local search (Step7-1)

Input:
S: the feasible schedule of FJSP.

begin
- **Step 1**: Identify a critical path P for a given incumbent solution S.
- **Step 2**: Set r to be the first operation of P. **repeat**
  - **Step 3**: Delete r from disjunctive graph $\mathcal{G}$ of S to get $\mathcal{G}^-$;
  - **Step 4**: Search for an assignable time interval assignable for r in $\mathcal{G}^-$;
  - **Step 5**: If no assignable interval for r is found, set r to be the next operation of P;
- **until** *(an assignable interval is found, allocate r in the interval. Otherwise, S is a local optimum of moving one operation)*;
- **Step 6**: go to **step7-2**.

end

Fig. 4.9 pseudo-code of critical path-based local search (Step7-1)

When moving a local optimum of operation, there exists no critical path P could be the mobile operations.That is to say, the interval did not assign any action can be found $r \in P$ which belongs to the disjunctive graph $\mathscr{G}^-$. Removing extra operations has brought more free time, meanwhile it may create time interval of negotiable for r, where the N_r is neighbors of operation r in the MN model. In this paper, we have developed VNS based local search algorithm. Algorithm 4.10 shows the basic process of moving two operations.

The idea of algorithm which based on neighborhood is straightforward: When moving a critical operation of local optimum obtained in step 7-1 is found, and we can try to improve the solutions by sequencing two operations determined in step 7-2 simultaneously, and at least one of them is important. If operations r and v are removed at the same time from $\mathscr{G}$ to get $\mathscr{G}^-$, (r is a critical operation) then we find the allocated time interval of r in $\mathscr{G}^-$. After that, find the allocated time interval of r in the same graph.

Variable neighborhood based local search (Step7-2)

Input:
S: the feasible schedule of FJSP.

begin
- **Step 1**: Identify a critical path P for a given incumbent solution S.
- **Step 2**: Set r to be the first operation of P.
- **repeat**
  - **repeat**
    - **Step 3**: Set v ($v \in N_r$) to be an operation in solution S.
    - **Step 4**: Delete r and the v from disjunctive graph $\mathcal{G}$ of S to get $\mathcal{G}^-$.
    - **Step 5**: Search for an interval assignable time for r in $\mathcal{G}^-$.
    - **Step 6**: If the assignable interval for r is found, insert r in the interval to get $\mathcal{G}^{-*}$; Otherwise, go to step 9.
    - **Step 7**: Search for an assignable time interval for v in $\mathcal{G}^{-*}$.
    - **Step 8**: If an assignable interval for v is found, insert v in the interval to get $\mathcal{G}^*$.
    - **Step 9**: If either the assignable interval for r or v is not found, set v to be the next operation in N_r.
  - **until** *the assignable intervals for both r and v are found or v is the last operation in N_r*;
  - **Step 10**: If the assignable interval for either r or v is not found, set r to be the next critical operation of P.
- **until** *both the assignable intervals for r and v are found; or r is the dummy terminating node*;

end

Fig. 4.10 pseudo-code of variable neighborhood based local search (Step7-2)

4.3 Experiments and discussion

There are many objectives can be considered in optimizing FJSP, for example, the total processing time, the total cost, the change of the workload, and so on.Among them, to minimize the makespan and often use is very important. Means manufacturing system can reduce time in limited time get the higher production efficiency.In this article, we consider the FJSP objective of minimizing makespan.

To investigate the performance of MREDA for different facets of problem difficulties, experiments are divided into five parts based on performance analyzing objectives.

(1) Computation Efficiency and Optimality on benchmark MK10 : Evaluating the computation times and convergence speed of the proposed MREDA and BEDA (Wang et al. [96]) respectively. In wang's research results, comparing BEDA and some existing algorithms prove the effectiveness of the proposed solving FJSP BEDA. Therefore, in this paper, We offer a wide range of research in computational benchmark problem, our approach is compared with the BEDA.

(2) Computation Efficiency on Benchmark set 3 benchmark sets: Evaluating the computation times of the proposed MREDA and BEDA respectively.

(3) Optimality on Benchmark set 3 benchmark sets: Evaluating the optimality of the proposed MREDA and BEDA respectively.

(4) Optimality Stability Analysis: Evaluating the dispersion of the proposed with BSEDA , BEDA respectively.

(5) Parameter tuning: The parameters used in MREDA are roughly divided into three types: learning MR, MR-based sampling and EDA control parameter such as truncation selection as promising solutions. In this paper, the maximum number of k-neighbors will be investigated.

As for FJSP, it should be noted that local search which based on critical path is incorporated into BEDA as well. In this section, exception for the experiment on analyzing the effectiveness of local search, MREDA refers to the proposal incorporating VNS local search without causing ambiguity.

The compared and proposed algorithms were all implemented by Eclipse using programming language Java, the machine environment is Intel I5 (2.3 GHz clock), 4G memory. Each result is the average value of 30 runs for all algorithms.

In order to offer the justice experiment environment to compare with other algorithms on the performance, the parameters and strategies used are shown in Table 4.1 respectively. The following several sets of benchmarks is taken into account.

(i) BRdata consists of 10 instances from Brandimarte [99].

(ii) DPdata is a set of 18 problems from Dauzère-Pérès [100].

(iii) BCdata consists of 21 instances from Barnes and Chambers [101].

Table 4.1 Parameters and strategies of BEDA and MREDA

	BEDA	MREDA
popSize	200	200
gen	2000	2000
Selection	-	-
Strategy	-	-
Operators	Sampling The critical path based on local search	Sampling Variable neighborhood based on local search
Parameters	$elitRate = 20\%$ $learingRate = 0.3$, 0.1 ter= 0.30	$elitRate = 20\%$ k = Max($\lceil Flex. \rceil$, 2) $\alpha = 1.5$ $coolingRate = 0.5$

4.3.1 Computation Efficiency and Optimality on benchmark MK10

To evaluate computation efficiency and optimality on one benchmark in detail. We conduct on benchmark MK10 whose #job is 20, #machine is 15, $Flex.$ 2.98, #operation is sampled from uniform distribution which minValue is 10, and maxValue is 15. and #processing timesis sampled from uniform distribution which minValue is 5, and maxValue is 20. Comparison of convergence Efficiency conducted on benchmark MK10 presents in Fig. 4.11.

For convergence, it presents that the proposal MREDA reduces computation costs (21%) than BEDA under reaching the same value of VTR (220). On the other hand, Our proposal achieves better optimality (improved 9%) than BEDA when iteration reaches 2000. While BSEDA takes more computation costs because of learning the Markov random Field.

4.3.2 Computation Efficiency

Calculate the cost based on evolutionary algorithm mainly depends on the number of health assessment.The time complexity of the difference between BEDA, our method mainly depends on the probability model.In particular, MREDA including two parts of Markov network: one is structure learning and the other is the conditional probability estimate using promising solutions.The process can be described in the following equations briefly:

$$\begin{aligned} \mathbf{COST} \propto\ & gen \times popSize \times \\ & |jobs| \times |operations| \times \\ & |machines| \times Flex. \end{aligned} \tag{4.6}$$

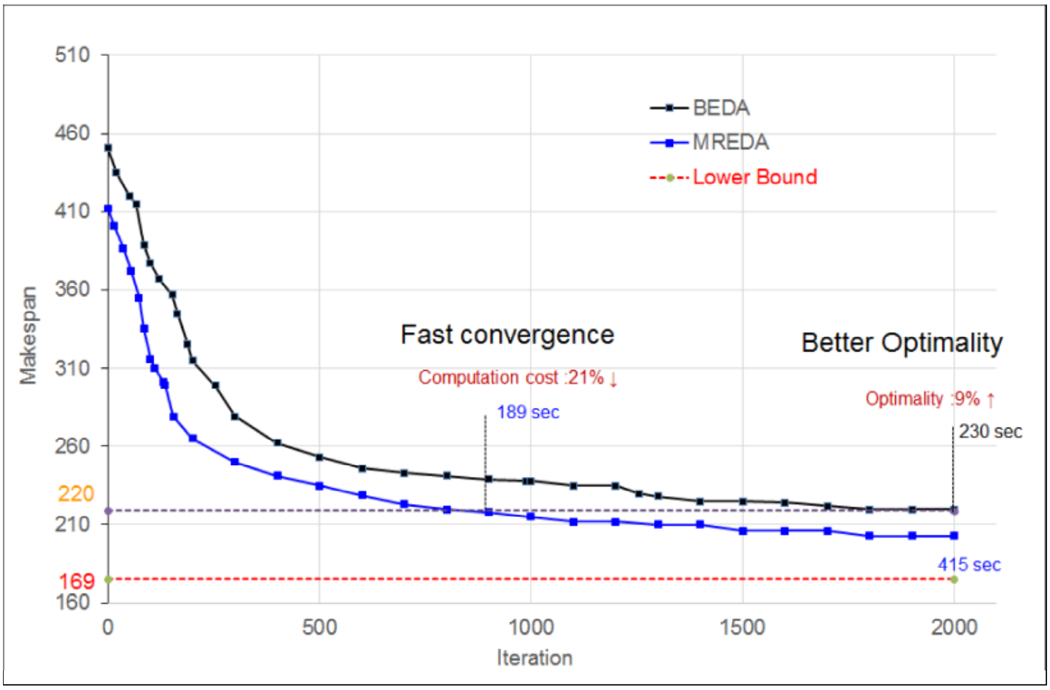
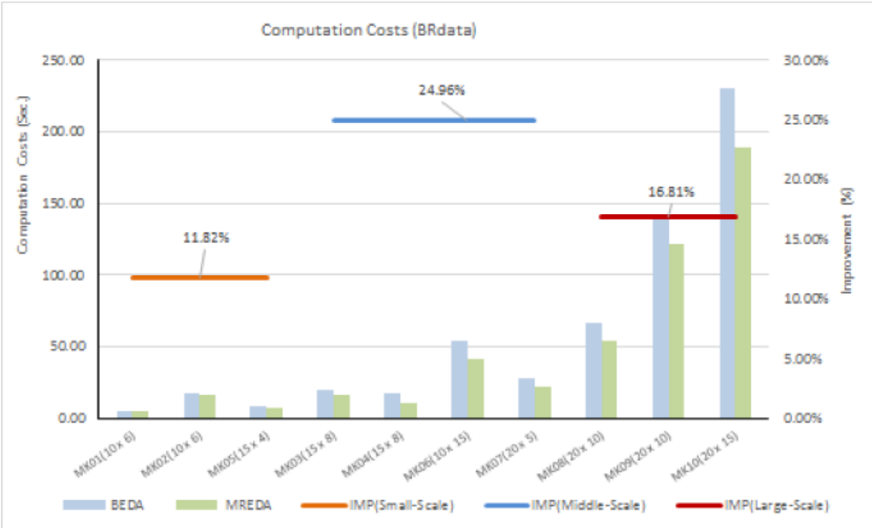


Fig. 4.11 Comparison of convergence Efficiency conducted on benchmark MK10

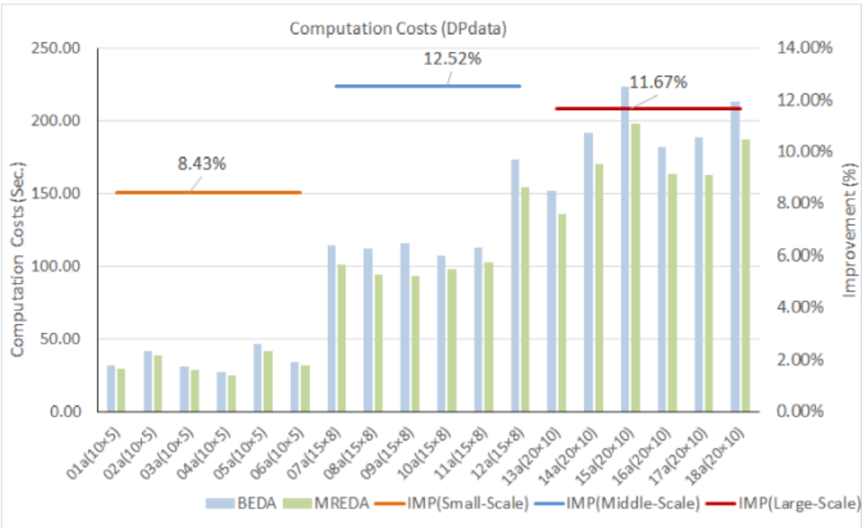
From equation 4.6, the complexity of structure learning not only decided by the parameters but also job numbers, the operation numbers, the machine numbers and the degree of Markov network. Fortunately, it is less than 10 in practice for degree of freedom..

First of all, we need to compare BEDA with MREDA in terms of the parameter settings shown in Table 4.1, and the termination criteria is the maximum number of iterations (generation). Table 4.2, Table 4.3 and Table 4.4 show the average of the results of the computation time, final solution was found in the experiments.According to the results, calculation level MREDA almost equal to the difference between MK0- MK06 BEDA level problem, however, when the problem complexity, than BEDA MREDA need more computational cost. Fig. 4.12 shows that the proposal achieve better computation efficiency than BEDA particularly middle-scale and large-scale problems.

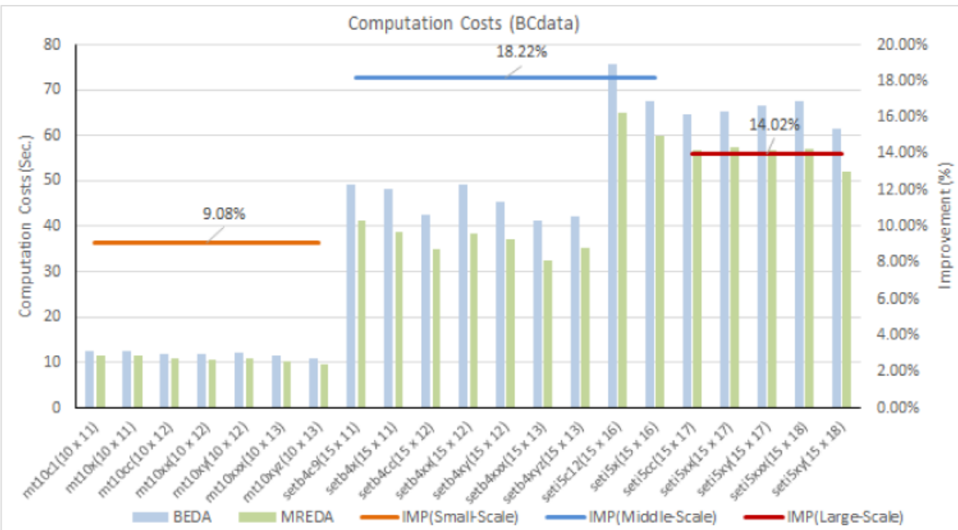
The second experiment is the BEDA convergence speed and our proposal by measuring the amount of the fitness evaluation (NFE) is the most commonly used measure in the literature in [102, 103]. Small NFE means higher convergence speed. Termination condition is to find a value less than the value-to-reach (VTR) reach the maximum number of iterations(*gen* given in Table 4.1). In order to compare the convergence speed, acceleration rate (AR) and success rate (SR) introduced by Rahnamayan et al [102] are involved in this study.



(a) Comparison of Computation Cost (BRdata)



(b) Comparison of Computation Cost (DPdata)



(c) Comparison of Computation Cost (BCdata)

Fig. 4.12 Comparison of Computation Cost by BEDA and MREDA conducted on 3 benchmark sets

Table 4.2 Comparison Result of computation times (sec.) of BEDA and MREDA conducted on BRdata

Problem	n x m	Flex.	BEDA	MREDA	Improvement(%)
MK01	10 x 6	2.09	5.11	4.41	13.6%
MK02	10 x 6	4.10	16.85	15.98	5.2%
MK03	15 x 8	3.01	19.57	16.58	15.3%
MK04	15 x 8	1.91	17.43	10.55	39.5%
MK05	15 x 4	1.71	8.64	7.20	16.7%
MK06	10 x 15	3.27	54.15	40.89	24.5%
MK07	20 x 5	2.83	27.66	21.96	20.6%
MK08	20 x 10	1.43	67.09	53.72	19.9%
MK09	20 x 10	2.53	139.22	121.49	12.7%
MK10	20 x 15	2.98	230.25	189.37	17.8%

Acceleration rate (AR) is based on the NFE for two algorithms MREDA and BEDA. It is defined as the following:

$$AR = \frac{NFE_{BEDA}}{NFE_{MREDA}} \tag{4.7}$$

where $AR > 1$ denotes that MREDA is faster. In this study, VTR of BRdata test cases is set to average makespan of MREDA (shown in Table 4.6) respectively.

Success rate (SR) measures the number of times for which the algorithm successfully reaches VTR for each test case. It is defined as the following:

$$SR = \frac{|\text{trials reaching VTR}|}{|\text{total trials}|} \tag{4.8}$$

where $|\text{trials reached VTR}|$ denotes the number of trials reaching the given VTR, and $|\text{total trials}|$ means total number of trials. The experiments are conducted under the condition where VTR of 14 test cases is set to best makespan of MREDA (shown in Table 4.6) respectively. Further, the average acceleration rate AR_{avg} and the average acceleration rate SR_{avg} over n test cases are calculated as the following:

Table 4.3 Comparison Result of computation times (sec.) of BEDA and MREDA conducted on DPdata

Problem	n x m	Flex.	BEDA	MREDA	Improvement(%)
01a	10×5	1.13	31.44	29.52	6.1%
02a	10×5	1.69	41.83	38.82	7.2%
03a	10×5	2.56	31.34	28.61	8.7%
04a	10×5	1.13	27.08	24.42	9.8%
05a	10×5	1.69	46.52	41.40	11.0%
06a	10×5	2.56	34.25	31.58	7.8%
07a	15×8	1.24	114.22	101.08	11.5%
08a	15×8	2.42	111.46	94.41	15.3%
09a	15×8	4.03	115.65	93.10	19.5%
10a	15×8	1.24	107.20	97.84	8.7%
11a	15×8	2.42	112.55	102.19	9.2%
12a	15×8	4.03	173.29	154.40	10.9%
13a	20×10	1.34	151.71	135.38	10.8%
14a	20×10	2.99	191.77	169.80	11.5%
15a	20×10	5.02	222.93	197.52	11.4%
16a	20×10	1.34	181.86	163.31	10.2%
17a	20×10	2.99	188.52	162.32	13.9%
18a	20×10	5.02	213.54	187.24	12.3%

$$AR_{avg} = \frac{1}{n}\sum_{i=1}^{n} AR_i \tag{4.9}$$

$$SR_{avg} = \frac{1}{n}\sum_{i=1}^{n} SR_i \tag{4.10}$$

The results of solving 10 test cases are given in Table 4.5. MREDA outperforms BEDA on all test problems in terms of AR. The AR_{avg} of the proposal is 1.38, which means the proposal is on average 38% faster than BEDA. For SR, the proposal achieves better results than BEDA. In particularly, BEDA fails to reach the VTR for the large scale problems (MK06, MK10).

Table 4.4 Comparison Result of computation times (sec.) of BEDA and MREDA conducted on BCdata

Problem	n x m	Flex.	BEDA	MREDA	Improvement(%)
mt10c1	10 × 11	1.1	12.33	11.57	6.2%
mt10cc	10 × 12	1.2	11.84	11.00	7.1%
mt10x	10 × 11	1.1	12.39	11.41	7.9%
mt10xx	10 × 12	1.2	11.79	10.52	10.8%
mt10xxx	10 × 13	1.3	11.49	10.36	9.9%
mt10xy	10 × 12	1.2	12.25	11.00	10.2%
mt10xyz	10 × 13	1.3	10.88	9.63	11.5%
setb4c9	15 × 11	1.1	49.05	41.35	15.7%
setb4cc	15 × 12	1.2	42.42	34.87	17.8%
setb4x	15 × 11	1.1	48.36	38.84	19.7%
setb4xx	15 × 12	1.2	49.32	38.27	22.4%
setb4xxx	15 × 13	1.3	41.28	32.44	21.4%
setb4xy	15 × 12	1.2	45.43	37.30	17.9%
setb4xyz	15 × 13	1.3	42.23	35.20	16.7%
seti5c12	15 × 16	1.07	75.72	64.97	14.2%
seti5cc	15 × 17	1.13	64.65	56.64	12.4%
seti5x	15 × 16	1.07	67.38	60.04	10.9%
seti5xx	15 × 17	1.13	65.32	57.28	12.3%
seti5xxx	15 × 18	1.2	67.49	56.96	15.6%
seti5xy	15 × 17	1.13	66.43	56.80	14.5%
seti5xy	15 × 18	1.2	61.52	52.11	15.3%

4.3.3 Optimality

Table 4.6, 4.7 and 4.8 gives the experiment results. Moreover, Fig. shows the improvement of optimality. From Table 4.6, 4.7 and 4.8 , it can be seen that MREDA performed better than BEDA in solving almost all the instances in term of of the best results. The small scale problems with lesser degree of MF like MK01, MK02, MK03, BEDA can obtain the best solutions under the best value of makespan. While the problem scale (MK06, MK10, 12a, 15a, 17a and 18a) increases, in particular for the problems with greater degree of machine flexibility, BEDA failed to find the near optimal. In BEDA, univariable probability distribution is used to model the probability distribution of machine assignment, that is, it

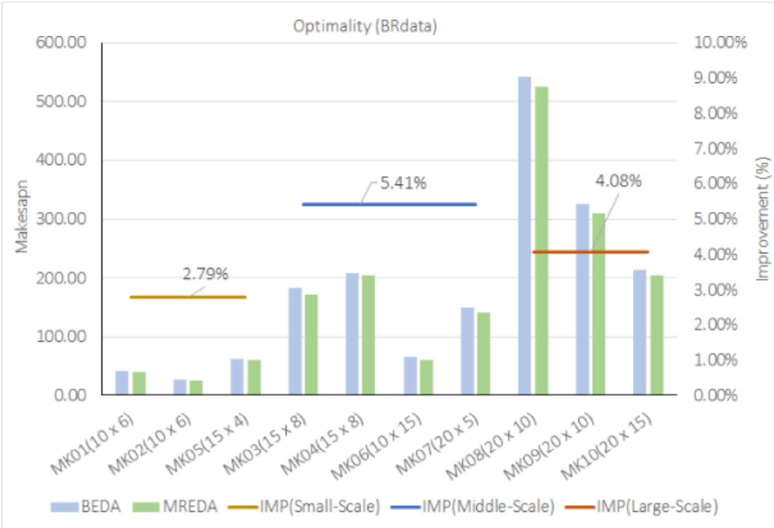
Table 4.5 Comparison of MREDA and BEDA on measurement AR and SR

Problem	n x m	BEDA		MREDA		*AR*
		NFE	*SR*	*NFE*	*SR*	
MK01	10 x 6	64530	0.67	41494	1	1.56
MK02	10 x 6	210660	0.56	152489	1	1.38
MK03	15 x 8	111908	1.00	76932	1	1.45
MK04	15 x 8	76730	0.64	110210	0.79	0.70
MK05	15 x 4	76786	0.69	50890	0.87	1.51
MK06	10 x 15	-	-	243904	0.85	-
MK07	20 x 5	160365	0.25	102763	0.85	1.56
MK08	20 x 10	243974	1.00	174408	1	1.40
MK09	20 x 10	387042	0.37	262151	0.65	1.38
MK10	20 x 15	-	-	244958	0.52	-
AVG			0.37		0.79	1.48

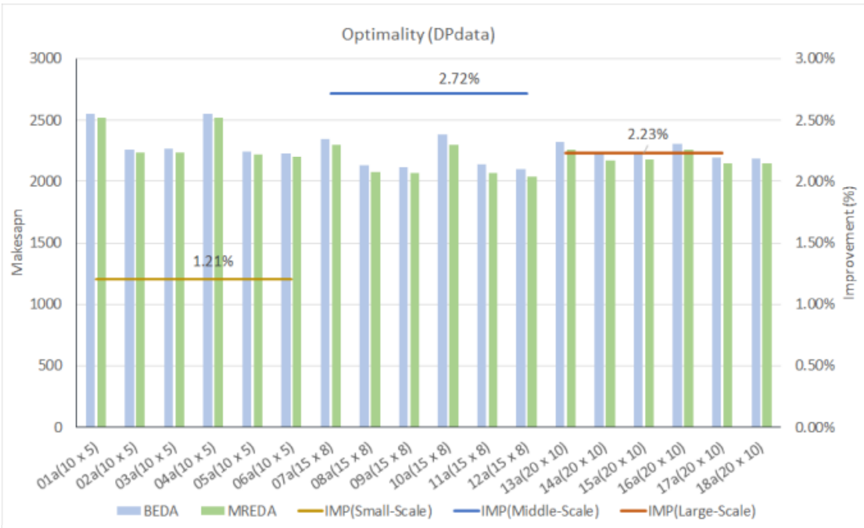
Table 4.6 Comparison of the proposed MREDA with BEDA on makespan (BRData)

Pro.	n x m	BEDA		MREDA	
		Best	**AVG**	**Best**	**AVG**
MK01	10 x 6	***40**	40.14	***40**	40.00
MK02	10 x 6	***26**	27.95	***26**	26.00
MK03	15 x 8	***204**	205.21	***204**	204.00
MK04	15 x 8	***60**	60.87	***60**	60.23
MK05	15 x 4	***172**	173.14	***172**	172.46
MK06	10 x 15	60	65.83	***58**	59.45
MK07	20 x 5	***139**	144.55	***139**	140.51
MK08	20 x 10	***523**	538.62	***523**	524.90
MK09	20 x 10	***307**	325.54	***307**	310.23
MK10	20 x 15	206	224.34	***200**	203.21

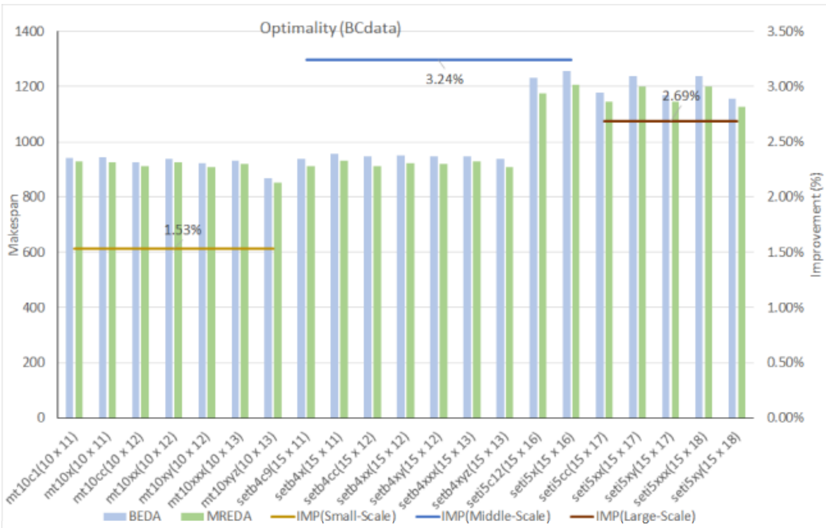
is assumed that the variable of machine assignment is independent. Unfortunately, BEDA does not consider potential interdependence between machine assignments of the operations which are assignable to the same machine. MREDA uses equation 4.5 to estimate the interdependence in order to adopt MR. MREDA variant provides prediction mechanism of decision variables for FJSP machine operation, and it achieves better stability than univariable probability distribution based algorithm like BEDA (shown in Fig. 4.15), although the effects of the accuracy of the prediction is a promising solution. For the best solution of Mk10, the Gantt chart and related decision information obtained by the MREDA is shown in Fig. 4.14, which is better than the result obtained by BEDA (makespan: 206). Moreover, our proposal achieves better performance than BEDA particularly for middle and large scale problems.



(a) Comparison of optimality (BRdata)



(b) Comparison of optimality (DPdata)



(c) Comparison of optimality (BCdata)

Fig. 4.13 Comparison of optimality by BEDA and MREDA conducted on 3 benchmark sets

Table 4.7 Comparison of the proposed MREDA with BEDA on makespan (DPdata)

Pro.	n x m	BEDA		MREDA	
		Best	AVG	Best	AVG
01a	10 x 5	***2518**	2548.72	***2518**	2518.35
02a	10 x 5	***2231**	2259.07	***2231**	2234.84
03a	10 x 5	***2229**	2268.22	***2229**	2239.65
04a	10 x 5	2515	2546.94	***2503**	2516.44
05a	10 x 5	2217	2246.54	***2216**	2218.32
06a	10 x 5	***2196**	2226.80	***2196**	2198.60
07a	15 x 8	2307	2345.77	***2283**	2297.62
08a	15 x 8	2073	2128.70	***2069**	2072.55
09a	15 x 8	***2066**	2112.93	***2066**	2066.69
10a	15 x 8	2315	2378.19	***2291**	2299.44
11a	15 x 8	2071	2136.72	***2063**	2067.81
12a	15 x 8	2038	2097.39	***2030**	2036.79
13a	20 x 10	2260	2322.58	***2257**	2260.66
14a	20 x 10	2171	2225.27	***2167**	2168.48
15a	20 x 10	2167	2222.10	***2165**	2172.45
16a	20 x 10	2258	2305.08	***2255**	2258.76
17a	20 x 10	2145	2190.44	***2140**	2146.77
18a	20 x 10	2130	2185.12	***2127**	2142.62

4.3.4 Stability Analysis

To evaluate the dispersion of MREDA, we compare BEDA and MREDA conducted on 12a, 15a, 17a and 18a problems of DPdata. The dispersion performances, as shown in Fig. 4.15 indicates that BSEDA is obviously better than BEDA, and BSEDA can also achieve satisfactory dispersion performance.

4.3.5 Parameter tuning

In MREDA, MR plays a crucial role to estimate the probability distribution. Generate the next sample solutions by the network created by estimate the probability distribution. In consideration of the length of the paper, discussion on experiment result analysis will concentrate only on the setting of the maximum number of k-neighbors.

For each node, to estimate the maximum number k of neighbors MR directly affects the balance between the model accuracy and the model complexity. Fig. 4.16 presents experiment result conducted on the problem 18a for $k = 2, k = 5, k = 9$, and the other parameters use the same parameters in Table 4.1. From Fig. 4.16, a higher k value may imply that the density

Table 4.8 Comparison of the proposed MREDA with BEDA on makespan (BCdata)

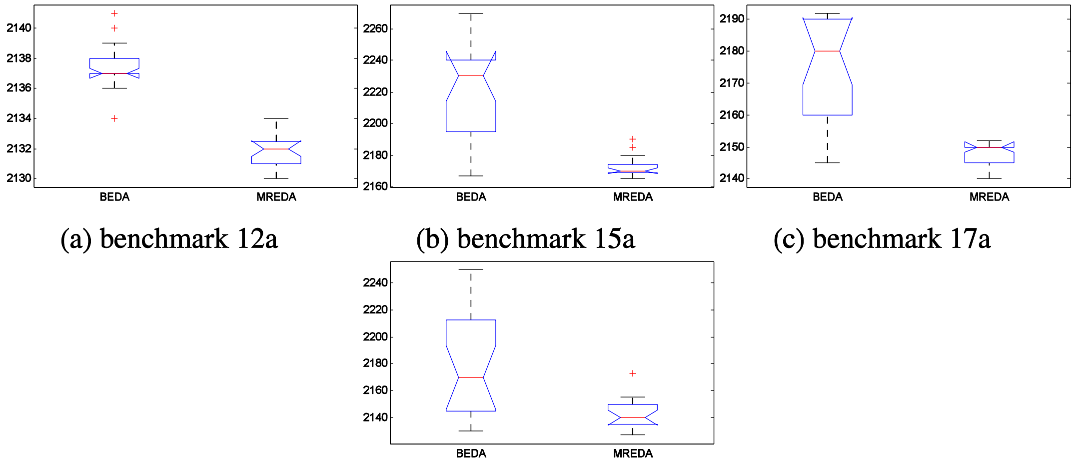
Pro.	n x m	BEDA		MREDA	
		Best	AVG	Best	AVG
mt10c1	10 x 11	928	941.76	***927**	928.11
mt10cc	10 x 12	*910	923.42	***910**	910.98
mt10x	10 x 11	*918	942.12	***918**	925.34
mt10xx	10 x 12	*918	936.23	***918**	922.90
mt10xxx	10 x 13	*918	931.02	***918**	919.15
mt10xy	10 x 12	906	921.43	***905**	907.45
mt10xyz	10 x 13	849	866.50	***847**	851.36
setb4c9	15 x 11	919	936.55	***914**	919.46
setb4cc	15 x 12	914	935.11	***909**	912.28
setb4x	15 x 11	*925	954.65	***925**	931.40
setb4xx	15 x 12	*925	948.61	***925**	926.14
setb4xxx	15 x 13	*925	945.85	***925**	926.77
setb4xy	15 x 12	*916	946.83	***916**	918.14
setb4xyz	15 x 13	*905	938.15	***905**	910.21
seti5c12	15 x 16	1175	1231.83	***1174**	1175.01
seti5cc	15 x 17	1138	1178.34	***1136**	1137.03
seti5x	15 x 16	1204	1256.52	***1201**	1200.51
seti5xx	15 x 17	1202	1247.06	***1199**	1200.71
seti5xxx	15 x 18	1204	1237.37	***1197**	1146.83
seti5xy	15 x 17	1136	1178.96	***1136**	1198.62
seti5xy	15 x 18	1126	1156.33	***1125**	1127.46

Fig. 4.14 The best solution of MK10 by MREDA (makespan = 200)

of interrelation between the decision variables increases, it may include excessive density of neighborhoods and potentially strive to raise some superfluous interrelation. Moreover, the MREDA with higher k value needs more computation costs for learning the structure of MR and decreases the speed of the convergence. On the other hand, a lower k shows that the high-order interrelation between the decision variables may be not encoded exactly in the MR model, which substantially increases the risk of inducing local premature convergence. Particularly, when set $k = 0$, the MR model degrades into univariable probability distribution based algorithms.

4.4 Summary

In this chapter, we study the application of MREDA to solve FJSP. In MREDA, Markov Random Field is involved to study the machine assignment according to the promising area of the search space. In addition, a critical path-based local search is presented to improve the performance of MREDA. A large number of experimental studies, and results confirm MREDA is superior to the recent research results on FMS scheduling problem. In the future work, we will improve the performance of MREDA, especially for more effective



(a) benchmark 12a (b) benchmark 15a (c) benchmark 17a

(d) benchmark 18a

Fig. 4.15 Boxplot of makespan by BEDA and MREDA conducted on 12a, 15a, 17a and 18a problems of DPdata

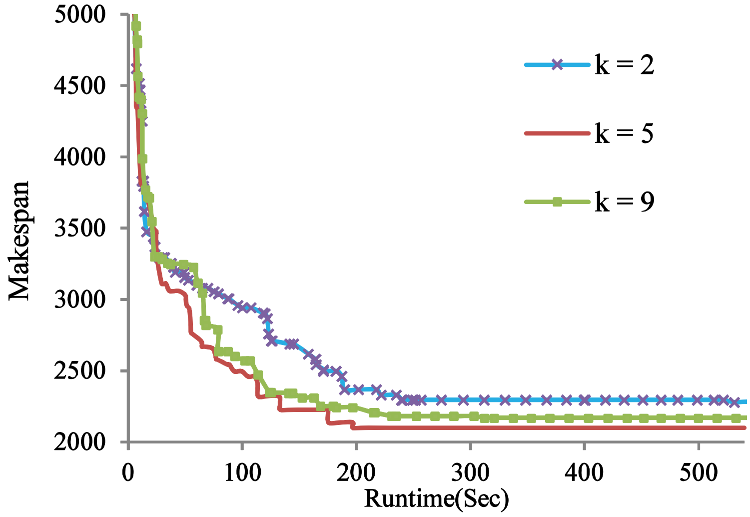


Fig. 4.16 Comparison of convergences on 18a with different *k*-neighbors

implementation structure learning algorithm. Furthermore, we should consider more effective Gibbs sampler variants of the algorithm constructing new solution. Although MREDA adopts an integrated approach for solving FJSP, it is crucial to analyze the effects of interdependent relationships existing between the machine assignments and the operation sequence.

Chapter 5

Bayesian network-based EDA (BSEDA) and its scheduling application

The main purpose of FJSP is to determine the process route of a job, while operations involved are allowed to any one of the multiple available machine candidates [83]. However, machine availability is the only constraint considered in the traditional FJSP. A more realistic scheduling model machine to other relevant resource constraints should be considered. In this study, we consider the FJSP (Type 2) having unlimited resource constraints on tools and tool approach directions (TADs) as found in some production systems, and changeover is taken into account. FJSP (Type 2) can be decomposed into two sub-problems: a process planning and a scheduling problem. The process planning is assigning each operation to a manufacturing resource (machine, tool) and determining the operation sequence according to the precedence relationship. The scheduling sub-problem is allocating all the selected resources in the work shop over the manufacturing time in order to obtain a feasible schedule with optimized production objectives [104]. FJSP considers the flexibility of each machine on the operation, and therefore expands search space for finding a scheduling solution. It is a more complex NP-hard problem and it incorporates all the difficulties and complexities of JSP [105]. Compared with classical FJSP, FJSP (Type 2) also takes into account the flexibility of resource utilization on the machines. These challenges significantly increase the complexity of FJSP (Type 2).

Previous researches showed there is a good performance when the traditional EAs are used for solving real application optimization problems. However the algorithm itself exists three characteristics lead to the traditional genetic algorithm (GA) is not entirely applicable for solving FJSP (Type 2). Firstly, the population of individual evolutionary convergence of the algorithm has a strong bias in response to a growing number of experiment area of the solution space allocation and higher than the average level of fitness [106]. Secondly, for FJSP (Type 2), there is a strongly interdependent relationship existing between the processing plan and the scheduling sequence of jobs. Unfortunately, the traditional EAs by the link problem of interaction between the decision variables of different hierarchical levels were inadequately analyzed. Finally, it could not be described as a single EA models due to the complex of problem and the strong constraints and goals from management entities.

Recently, there are growing interests in EDAs that establish a clear good solution of distribution probability model found so far, using structural model to guide the search performance further [107]. In BSEDA, Bayesian network is employed in modeling the decision variables for estimating the joint distribution, and the Bayesian network is constructed from a chosen metric and various constraints in each generation. When prior knowledge about the relationship of the variables is not known, structure learning of Bayesian networks is NP-hard problem [108]. Consequently, in solving the practical problem, BSEDA will require long computation time to obtain the near optimal solution.

This chapter presents Bayesian network based EDA (BSEDA). In BSEDA, Bayesian network is employed to model the joint distribution of the multi-dimensional variables. It provides a prediction mechanism of parenthood on the variation of decision variables, and experiment shows that it attains better steadiness compared with random search strategy, and the accuracy of the structure and parameters of Bayesian network is affected by the promising solutions. Presented BSEDA can be actually applied to IFJSP (Type 2). Different with FJSP (Type 1), FJSP (Type 2) considers manufacturing environment with multi-purpose machines equipped with tool-box, and the tool changeover and job changeover have highly depend on the processing sequence (sequence-depended). Bayesian network is employed to learn machine assignment of the operations affecting the make-span, simultaneously model the casual-relationship between tools equipped on machine.

Furthermore, the cooperative BSEDA is studied where the cooperative co-evolutionary mechanism is used to analyze the features of the original model, meanwhile to reduce the structure learning time of BSEDA by separating the problem into several sub-problems on the real separation of managing responsibilities. The decision space of each sub-problem is encoded into representation of a sub-population (species). The species is evolved in its own population and adapts to the environment through the evolutionary policy iteration of an EA. In evolution, species interaction with predefined generation time interval, to ensure that the ecosystem within the adaptability of the total population. For each species, Bayesian network is employed in modeling the decision variables for estimating the joint distribution of multinomial data in order to generate new solutions. The proposed algorithm is not only capable of maintaining searching diversity in the evolution, but also incrementally estimates the distribution of the decision variable s from the promising data.

5.1 Efficient Bayesian Network Learning Algorithm

5.1.1 Structure learning

In the learning community, learning Bayesian network has been proved that to be NP-hard, particularly it is hard to find the optimal when prior knowledge about the relationship between the variables is not known [109]. In the context of BSEDA, learning the PGM is to study interactions between the variables based on a performance metric, rather than attempt to find the optimal model. Consequently, A score-and-search approximate approach is developed to find a PGM that encodes all significant interactions in a problem.

In graph theory, the *skeleton* of DAG G is the undirected graph resulting from removing all of the arrowheads from the DAG G . Therefore, learning Bayesian network structure can be considered to be the search paradigm by double application. First, we construct a skeleton of DAG G using the over given data D. In the exposition above, learning Markov Network in section 4.1.1 presents the process of building undirected graph model from the given data. It is worth noting that the assumption is made that the distribution of observed variables is faithful to a graph model. Second, Starting from the skeleton, greedy algorithm performs arrowhead addition that improves the quality of current network the most given measure criterion. The algorithm 5.1 shows the pseudo-code of greedy algorithm illustrated above.

In order to evaluate the score a model, we involve Bayesian information criterion which is often used as measurement in learning community. Let r_i denote the number of states for variables X_i, and the number of configurations over the parents for the variable X_i in G is defined as $q_i = \prod_{X_l \in \Pi_i} r_l$. The BIC is calculated as the following equation:

$$BIC(G|D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \mathscr{N}_{ijk} \log_2 (\frac{\mathscr{N}_{ijk}}{\mathscr{N}_{ij}} - \frac{\log_2 \mathscr{N}}{2} \sum_{i=1}^{n} q_i (r_i - 1) \tag{5.1}$$

where $\mathscr{N}_{ijk}$ denotes the number cases in the promising data D with X_i in its kth configuration and Π_i in the jth configuration. It should be noted that we let $q_i = 1$ if the variable X_i doses not have any parents.

For example, the structure consists of two variable $x_1 \in \{yes, no\}$ and $x_2 \in \{pos, neg\}$. The promising data D is given as described Fig. 5.2. We illustrate how to calculate the

```
Step5: Estimating structure of Bayesian network
Input:
  X: X = (X_1, X_2, ..., X_n), vector of the decision variables.
  D: promising solutions determined by Step 3.
  E_op: elementary opeations set: {Direction addtion, Direction Reversal}.
  score(G): scoring function that measures the quality of G.
  Edges(S): the set of edges given skeleton S.
Output:
  G: Constructed Bayesian network
begin
    Step 4.1:Construct the skeleton S of Bayesian network G.
    Step 4.2: G ← φ.
    for e ∈ Edges(S) do
        Step 4.3: performs elemetary operation in E_op to improve the score(G). begin
            for op ∈ E_op do
                G' ← op(G, e).
                if score(G') > score(G) then
                    G ← G'.
                end
            end
        end
    end
end
```

Fig. 5.1 pseudo-code of estimating structure of Bayesian network

$BIC(x_1 \to x_2|D)$ which means the variable x_2 is dependent on the variable x_1. the value of BIC can calculated as follows:

$$BIC(G|D)$$
$$= \big[8 \cdot \log\left(\tfrac{8}{8+2}\right) + 2 \cdot \log\left(\tfrac{2}{8+2}\right) +$$
$$6 \cdot \log\left(\tfrac{6}{6+2}\right) + 2 \cdot \log\left(\tfrac{2}{6+2}\right) +$$
$$0 \cdot \log\left(\tfrac{0}{0+2}\right) + 2 \cdot \log\left(\tfrac{2}{0+2}\right)\big] - \tfrac{\log(10)}{2} \cdot (1 \times (2-1) + 2 \times (2-1))$$

$= -18.69$ Similarly, we can calculate $BIC(x_2 \to x_1|D)$.

5.1.2 Parameter estimation

Assume that the structure of BN is learned over the variables X by Step 5, But the conditional probabilities should be estimated. That is, parameter estimation of the model should be done on the given the promising data D. If D is a data set of samples, where each sample is a configuration over all the variables in the vector X, Then such a case is called *complete data*. Within the learning field, a parameter is typically denoted by θ, and the estimation of

Case(#)	X_1	X_2
1.	*yes*	*pos*
2.	*yes*	*pos*
3.	*yes*	*pos*
4.	*yes*	*pos*
5.	*yes*	*pos*
6.	*yes*	*pos*
7.	*yes*	*neg*
8.	*yes*	*neg*
9.	*no*	*neg*
10.	*no*	*neg*

$X_1 \rightarrow X_2$

X_1	yes	no
	8	2

		X_1	
		yes	*no*
X_2	*pos*	6	0
	neg	2	2

Fig. 5.2 Example of promising data for Bayesian network learning

parameter is denoted by $\hat{\theta}$. Let B_θ be the BN with respect to the parameters θ, maximum likelihood estimation is broadly used to choose a parameter $\hat{\theta}$ that maximizes the likelihood:

$$\hat{\theta} = \arg\max_{\theta} L(M_\theta|D) \tag{5.2}$$

where $L(M_\theta|D) = \prod_{d\in D} P(d|B_\theta)$ is called the likelihood of B_θ given each case $d \in D$. It should be noted that each case $d \in D$ is assumed to be independent given the model. In other hands, for each conditional probability distribution, frequencies given the data D is used as estimates to achieve a maximum likelihood estimate. For example, the simple calculation is defined as follows:

$$p(x|y) = \frac{\mathcal{N}(X = x, Y = y)}{\mathcal{N}(Y = y)} \tag{5.3}$$

where $\mathcal{N}(X = x, Y = y)$ represents the size of instances of D with the variable $X = x$ and $Y = y$, and $\mathcal{N}(Y = y)$ represents the size of instances in the set D with the variable $Y = y$.

5.1.3 Sampling

After the structure and conditional probabilities tables (parameter) of Bayesian network G have been learned, new candidate solution can be sampled on the distribution encoded in the learned PGM (in equation 2.14). Bayesian network can be considered as casual-network [110]. Inherently, a new sample can be generated using the probabilistic logic , which aims to determine the value of decision variables following the ancestral ordering of the nodes.

Therefore, it is vital to determine the ancestral ordering of nodes. That is, for each variable in generated variable decision sequence, the values of its parent should be decided before it. The Fig. 5.3 shows the pseudo-code of probabilistic logic sampling process.

Step7: Probabilistic Logic Sampling based on Bayesian network

Input:
$X = (X_1, X_2, \ldots, X_n)$: vector of the decision variables;
$popSize$: number of the population solutions;
D: promising solutions determined;
G: estimated structure of Bayesian network;
Π: a Parenthoods system on X, where each N_i is the set of variables that is parent of the variable X_i;
Output:
$candidates$: set of new solutions created by Probabilistic Logic Sampling;

```
begin
    for j ← 1 to popSize do
        step 7.1: N^(G) ← the set of nodes that have no parents in G ;
        while N^(G) ≠ ∅ do
            step 7.2: Choose a variable X_i from N^(G) randomly;
            step 7.3: Using the selected set of solutions D, estimate the conditional
            p(x_i|Π_i) for each value x_i of the variable X_i;
            step 7.4: Sample new x_i according to the conditional probability
            distribution P(x_i|Π_i);
            step 7.5: N^(G) ← N^(G)\{X_i}, for each variable X_i' of the children of
            X_i, if the value of each variable in N_i' is decided, N^(G) ← N^(G) ∪ {X_i'};
        end
    end
end
```

Fig. 5.3 pseudo-code of sampling based on probabilistic logic

5.2 Design for FJSP (Type 2)

The following subparagraph presents the vital components of BSEDA which the procedure is shown in Fig.3.5 in detail. Primarily, how to encode in Bayesian network is described. Then and there, the new alternative solution is inferred on the Bayesian network. To improve the optimality, the Bayesian network based local search algorithm is developed. Lastly, GA-based local search strategy is taken into account to enhance the exploitation capability and maintaining certain population diversity.

5.2.1 Bayesian network-based Encoding

The following subsection presents the proposed BSEDA for FJSP (Type 2) in detail. Firstly, we describe the structure of Bayesian networks, learning of machine assignment and tool setup based on the Bayesian network. Then, we present how each individual exploits solution space based on the Bayesian network. Finally, we summarized the brief introduction of cooperation mechanism and formulation of BSEDA.

For FJSP (Type 2), it can be divided into three problems: process planning, operation sequencing, and jobs scheduling. The effect of job scheduling is that determine the processing order of operations for schedule S.

Most of the literatures always determine the operation of the machine and related resources, tools and TADs) to limit work. Since then, the operation sequence S structure according to the decisions in front of the machine and resources, it can be written as

$$D(X_{ij}) \rightarrow D(Y_{ij}) \rightarrow D(Z_{ij}) \rightarrow S \tag{5.4}$$

where $D(X_{ij}), D(Y_{ij}), D(Z_{ij})$ denotes the machine assignment, tools selection, and TADs selection of operation o_{ij} respectively.

Unfortunately, decisions on tools and TADs selections are crucially influenced by the immediately preceding resources decisions on the same machine; it is referred to sequence-dependent [82]. It increases the potential of changeover when the sequence-dependent constraint is not sufficiently considered ahead. Therefore, in this paper, we adopt a novel approach for FJSP (Type 2) with the following sequence:

$$S \rightarrow D(X_{ij}) \rightarrow D(Y_{ij}) \rightarrow D(Z_{ij}) \tag{5.5}$$

The construction algorithm of operation sequence S is not a major concerning factor of our approach. In fact, any meta-heuristic algorithm (GAs, permutation-algorithms) could be adopted. The Bayesian network of machine assignments, tools selections and TADs selections are mainly discussed.

The Bayesian network of machine assignment for each job adopts the cyclic directed graph method to encode the solution structure of the problem. The node represents the discrete variables X_{ij} and the directed edge between two nodes shows the relationship between adjacent operations.

Therefore, the parents of variables X_{ij} in operation sequence S largely decide the decision of X_{ij}. The conditional probability of X_{ij} being assigned to machine m is defined as:

$$p(x_{ijm} = 1) \left| \Pi^M \ (X_{ij}, S) \right) = \frac{\mathrm{N}(x_{ijm} = 1, \pi^M(X_{ij}, S))}{\sum\limits_{m' \in X_{ij}} \mathrm{N}(x_{ijm'}, \pi^M(X_{ij}, S))} \tag{5.6}$$

where $\Pi^M(X_{ij}, S)$ represents the machine m on which immediately preceding operation of operation o_{ij} in sequence S was carried out, and these two operations belong to the same job i. $\mathrm{N}(x_{ijm} = 1, \pi^M(X_{ij}, S))$ denotes the number of instances in promising solutions with variable x_{ijm} = 1 (that is $X_{ij} = m$) and the machine of preceding operation within job $\Pi^M(X_{ij}, S) = \pi^M(X_{ij}, S)$.

Bayesian networks of tool selection for each job uses hierarchical acyclic directed graph method. It is used to encode the problems of solution structure.The nodes in the Bayesian network denote the discrete variable Y_{ij}, referring the alternative machines that can be configured with the tool Y_{ij}. The relationship between the directed edge represents the specify alternative tools for machines and tools Y_{ij} of operation o_{ij}. So, the conditional probability of Y_{ij} with respect to X_{ij} in the operation sequence S can be written as :

$$p(y_{ijl} = 1) \left| \Pi^T \ (Y_{ij}, S) \right) = \frac{\mathrm{N}(y_{ijl} = 1, \pi^T(Y_{ij}, S))}{\sum\limits_{l' \in Y_{ij}} \mathrm{N}(y_{ijl'}, \pi^T(Y_{ij}, S))} \tag{5.7}$$

where $\Pi^T(Y_{ij}, S)$ represents the tool which was setup for the machine determined by x_{ijm} in sequence S, denotes the number of instances in promising solutions with variable $y_{ijm} = 1$ (that is $Y_{ij} = l$) and the preceding tool.

Similarly, for the TAD selection, the nodes in the Bayesian network denote the discrete variable Z_{ij}, referring the alternative TAD that can be configured with the tool Y_{ij}. The conditional probability of Z_{ij} with respect to X_{ij} in the operation sequence S. can be presented as:

$$p(z_{ijd} = 1) \left| \Pi^D \ (Z_{ij}, S) \right) = \frac{\mathrm{N}(z_{ijd} = 1, \pi^D(Z_{ij}, S))}{\sum\limits_{d' \in Z_{ij}} \mathrm{N}(z_{ijd'}, \pi^D(Z_{ij}, S))} \tag{5.8}$$

where $\Pi^T(Z_{ij}, S)$ shows the TAD selection for the machine chosen by x_{ijm} in sequence S, N denotes the instance number in promising solutions with variable $z_{ijd} = 1$ (that is $Z_{ij} = d$) and the preceding TAD.

Equations 5.6, 5.7, 5.8 define the likelihood of machine changeover between the operations and between the resources on a machine respectively, however, it is not sufficient for Bayesian inference. The marginal probability of the prior element is required; equation 5.9 defines the marginal probability of the random decision variables X_{ij}, Y_{ij} and Z_{ij}:

$$\begin{aligned}
p(x_{ijm} = 1) &= \frac{\mathrm{N}(x_{ijm}=1)}{\sum\limits_{m' \in X_{ij}} \mathrm{N}(x_{ijm'}))} \\
p(y_{ijl} = 1) &= \frac{\mathrm{N}(y_{ijl}=1)}{\sum\limits_{l' \in Y_{ij}} \mathrm{N}(y_{ijl'}))} \\
p(z_{ijd} = 1) &= \frac{\mathrm{N}(z_{ijd}=1)}{\sum\limits_{d' \in Z_{ij}} \mathrm{N}(z_{ijd'}))}
\end{aligned} \tag{5.9}$$

We introduce one illustrative example to demonstrate the process of Bayesian network structure learning for machine assignment, which is shown in Fig. 5.4: Initially, we have the knowledge of job routing information of two jobs: Job 1 and Job 2, in which there are 4 operations and 3 operations respectively. In step 1, we initialize the Bayesian network with 7 nodes which has no connections between each pair nodes. Each node X_{ij} represents the decision variable of machine assignment for operation O_{ij}. Next step, we construct the skeleton by mutual information (MI) by Equation 4.1 . Here we can get a network without direction. Then, if there are two operations have edge in job routing, but the corresponding nodes in Bayesian network have no connection after MI calculation, we add one edge between them. When two operations have precedence relationship, it is obviously these two nodes have strong relationships due to machine setup time or resource constraints, so that in Bayesian network, we have to add one edge to represent the parenthood relation. Thereafter, the direction of edge should be decided in step 4. One simple way is to calculate the value of Bayesian Information Criterion according to Equation 5.1 to generate the direction, starting from a node randomly selected. In order to make the process much more stability, we utilize the precedence relations in job routing. As a result, we generate the direction in Bayesian network based on the directions in job routing. In step 5, based on the direction generated in step 4, we calculated the BIC to decide the direction of the edges which do not solved in step 4. Finally, the Bayesian network is structured after we check whether there is cyclic in the network we have just built.

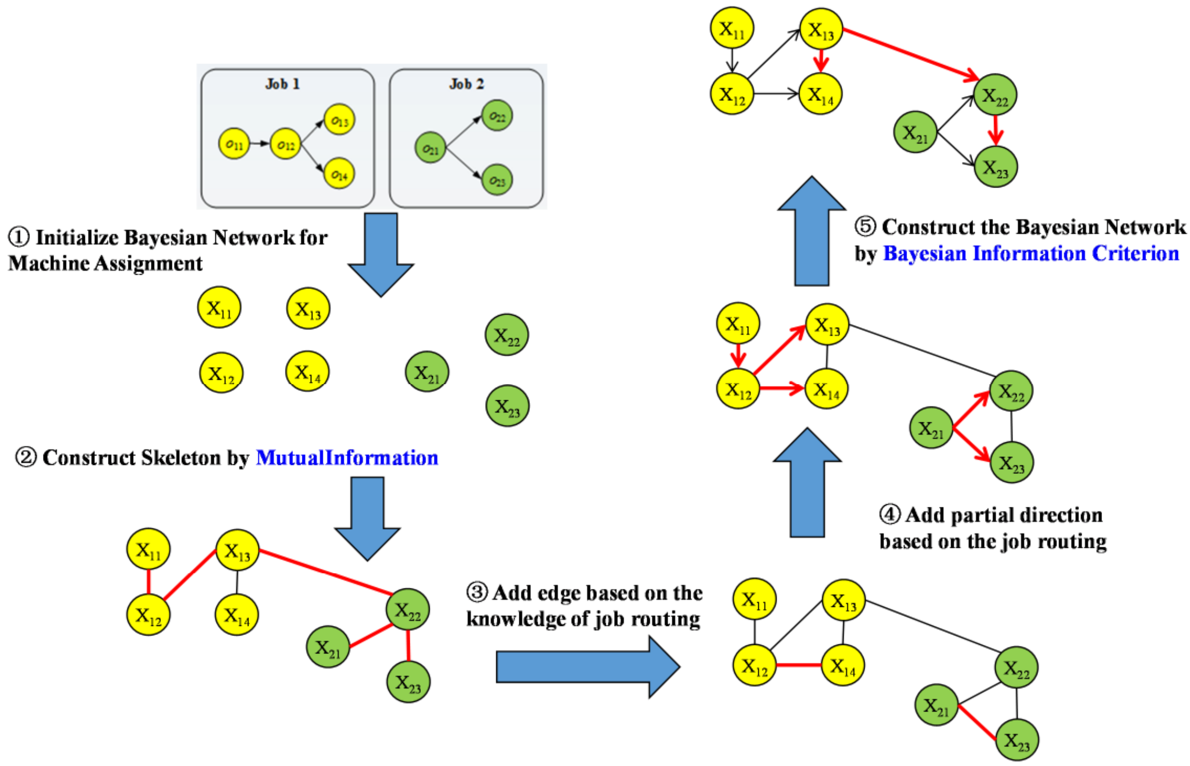


Fig. 5.4 Illustration of Bayesian network learning

5.2.2 LocalSearch

The local search is provided by a suitable hybridization using problem-specific solvers including heuristics, approximate, and exact algorithms. Essentially, local search algorithms try to find high quality solutions by searching the solution space. More precisely, these algorithms start with an initial solution, and then iteratively generate a neighboring solution in terms of small changes that may be applied to a solution.

BSEDA pay more attention to global exploration, while it exploitation capability is limited, Hence, the trade-off between the exploration and the exploitation should be taken into account in BSEDA. In the meantime, it often appears the stagnation in the EDA after running some iterations. It is a prevalent problem for the EDA that entirely samples new solutions by using the probability models because of losing the diversity [111]. In order to achieve considerable optimality, mutation operator-based GA local search strategy is employed to enhance the exploitation capability and maintaining certain population diversity.

Since FJSP (Type 2) consists of two-subproblems, i.e., operation sequencing and resource allocation, it may optimize the objective function by adjusting the operation sequence and the resource allocation,respectively. For local exploitation, it is not necessary to perturb both operation sequence and resource allocation simultaneously because of exploiting neighborhood decision space.

if the best solution found so far is unchanged at consecutive iteration ter , the alternative solutions generated by BSEDA will be cloned into two subpopulations to generate neighbor individuals with different methods. we employ GA local search strategy to adjust the operation sequencing and resource allocation respectively. Fig. 5.5 shows the framework of GA-based Local search incorporating into BSEDA.

Operation sequence adjustment operator for Sub-P1

The sub-population Sub-P1 is only to adjust the operation sequence. So, We apply order crossover for the operation sequence vectors. Briefly, The order crossover procedure selects a subsection of operation sequence from one parent at random, Then, it produces a proto-child by copying the substring of operation sequence into the corresponding positions. Finally, It place the operations into the unfixed positions of the proto-child from left to right according to the order of the sequence in the second parent. Illustration of order crossover is shown in Fig. 5.6.

Resource allocation adjustment operator for Sub-P2

In this dissertation, we also consider critical-path local search for scheduling problem which is always taken into account in other research. For FJSP (Type 2), once the parameters of Bayesian network for machine assignment and tool setup have been learned from the promising solutions respectively, that is, the relationship between adjacent operations with the same job and the tool changeover on the machine can be estimated. While variant operator is performed on one operation o_{ij}, the joint probability on machine assignment of o_{ij} with its predecessors should be compared with each other, and the predecessor $pre(o_{ij})$ with maximum probability is selected; if the probability is identical, $pre(o_{ij})$ is selected randomly. Therefore, the operation o_{ij} is assigned to machine m on which the operation $pre(o_{ij})$ has been assigned. It must be decided now where would be the reasonable place to insert o_{ij}. Similarly, the joint probability on tool setup of o_{ij} with its operations following $pre(o_{ij})$ will be compared, and the operation with maximum probability is selected and o_{ij} will be

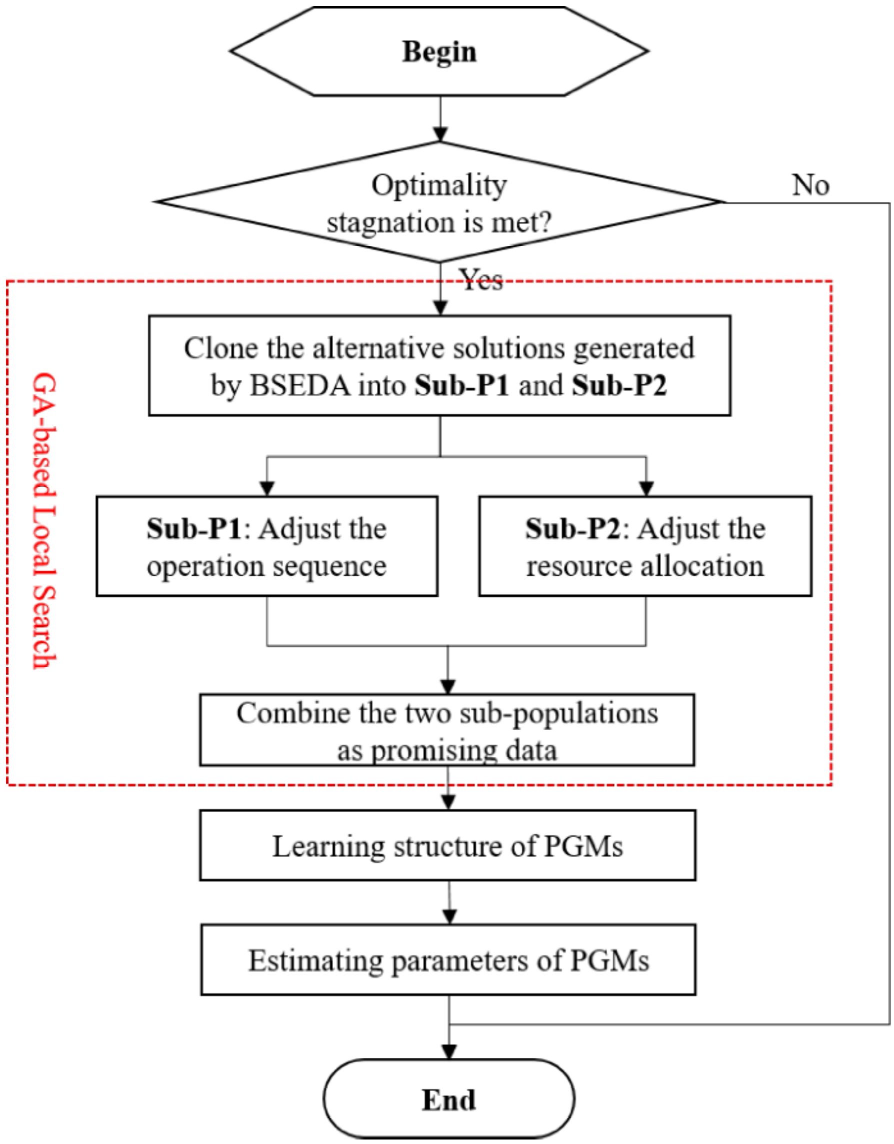


Fig. 5.5 The framework of GA-based Local search incorporating into BSEDA

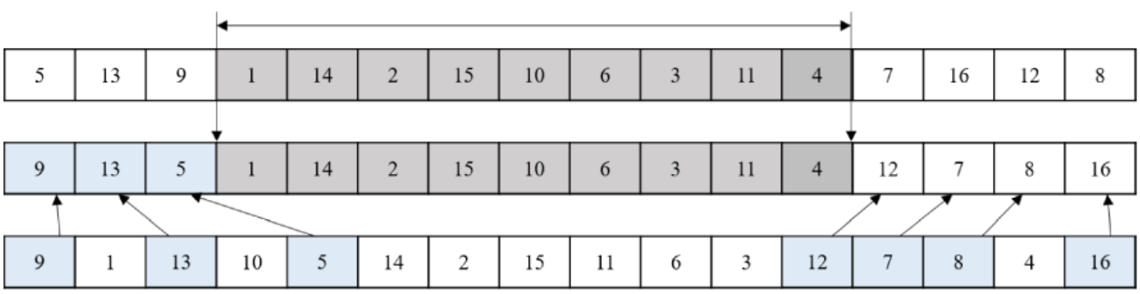


Fig. 5.6 Order crossover procedure’s illustration for operation sequence

inserted after it. Fig. 5.7 shows the pseudo-code of evolutionary operator on individual for FJSP (Type 2).

a) Illustration of learning Bayesian-network: For each job species, Bayesian network is used to model joint probability between adjacent operations regarding machine assignment for the same job. For example, Fig. 5.8 shows the Bayesian model of machine assignment (X_{ij}) for job 1. The solid directed line represents the physical constraint on machining order, that is, o_{12}, o_{13} cannot be performed before o_{11}. The dotted-line denotes that either operation o_{12}, o_{13} can be performed after operation o_{11} is machined completely. Table 5.1 presents joint probability table about o_{12} of job 1 on machine assignment. The matrices of conditional probability tables of Bayesian network is constructed, and the values are set with the same value initially. When the ga-based local searcher collects alternative solutions and selects 100 better solutions as promising data, if $\alpha = 0.5$, the values of joint probability table should be updated according to equation 5.6. The marginal probability of X_{12} is estimated by equation 5.9. Similarly, the resource selection about the tool and TAD on the machine can be calculated according to equations 5.7, 5.8 and 5.9.

Evolutionary operator of GA-based Local search for IPPS

Input:
S: operation sequence including all jobs include in partnership;
$rand()$: the random double number and $rand() \in [0, 1)$;
$\hat{T}[m]$: Index of the tool that was setup to machine m;
$\hat{D}[m]$: Index of the TAD that was setup to machine m;
$\hat{M}[i]$: Index of the machine on which the job i was machined recently;
i : Index of the job that exploiting the machine assignment and scheduling;
p_i : The probability of change the decision variable;
Output:
v: new solution.

begin
  for $o_{ij} \leftarrow S(i)$ **do**
    if $(rand() < p_i)$ **continue**;
    $\pi^M(o_{ij})' \leftarrow \{\pi^M(o_{ij}) : \max(p(x_{ijm} = 1) \,|\, \pi^M\ (X_{ij}, S)) \times p(x_{ijm} = 1)\};$
    if $|\pi^M(o_{ij})'| > 1$ **then**
    $pre(o_{ij}) \leftarrow \pi^M(o_{ij})'[1];$
    else $pre(o_{ij}) \leftarrow select\ \pi^M(o_{ij})'$ at $random$;
    end
    $m' \leftarrow machine\ on\ pre(o_{ij});$
    $\pi^D(o_{ij})' \leftarrow \{\pi^D(o_{ij} : \max(p(z_{ijl} = 1) \,|\, \pi^D\ (Z_{ij}, S) = \hat{D}[m']) \times p(z_{ijl} = 1)\};$
    $\pi^T(o_{ij})' \leftarrow \{\pi^T(o_{ij}) : \max(p(y_{ijl} = 1) \,|\, \pi^T\ (Y_{ij}, S) = \hat{T}[m']) \times p(y_{ijl} = 1)\}$
    if $|\pi^D(o_{ij})')| > 1$ **then** $d' \leftarrow select\ \pi^D(o_{ij})'\ at\ random;$
    $l' \leftarrow \pi^T(o_{ij})'[1]$ *or* **if** *mutiple at random*;
    else
    $d' \leftarrow \pi^D(o_{ij})'[1];$
    $l' \leftarrow \pi^T(o_{ij})'[1]$ *or* **if** *mutiple at random*;
    end
    $x_{ijm'} = 1;$
    $y_{ijl'} = 1;$
    $z_{ijd'} = 1;$
  end
  Evaluates the partnership;
  output new alternative solution;
end

Fig. 5.7 pseudo-code of evolutionary operator of BSEDA for FJSP (Type 2)

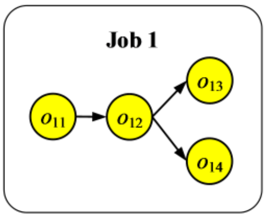


Fig. 5.8 Structure of Bayesian-network for job 1

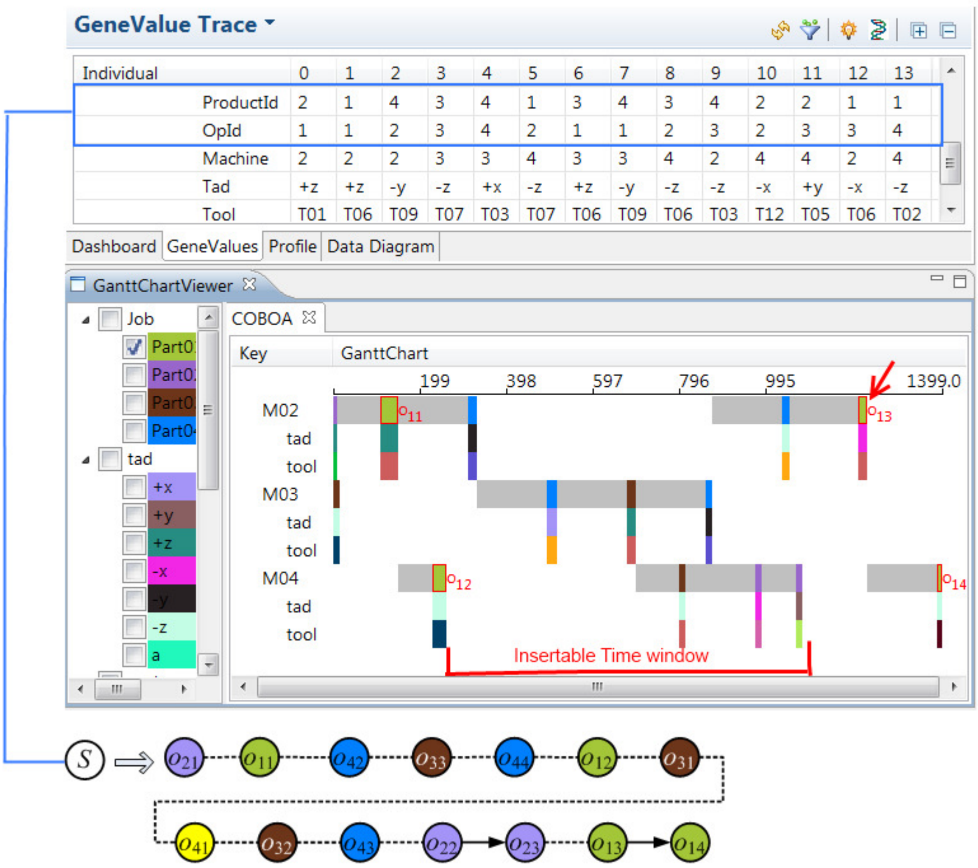


Fig. 5.9 Illustration of Bayesian-based evolutionary operator

Table 5.1 Joint probability table about machine assignment related to o_{12} of job 1

PNode	Machine on PNode	CNode	Machine on CNode	Joint probability	Number of promising data	New Joint probability
o_{12}	m_3	o_{13}	m2	0.5	10	0.58
	m_3	o_{13}	m4	0.5	5	0.42
	m_3	o_{14}	m2	0.5	10	0.58
	m_3	o_{14}	m4	0.5	5	0.42
	m_4	o_{13}	m2	0.5	10	0.38
	m_4	o_{13}	m4	0.5	30	0.63
	m_4	o_{14}	m2	0.5	5	0.33
	m_4	o_{14}	m4	0.5	25	0.67

$\alpha = 0.5$ $P(x123) = 0.3$ $P(x124) = 0.7$

b) Illustration of Bayesian-based evolutionary operator: After the initialization of BSEDA as shown in Fig. 3.5 is completed. This presents a search point in the decision space for the determination of the operation schedule. Sub-P2 starts evolving iteratively in which evolutionary operator is performed on individual under the probability p_i. We will examine closely into the process of the proposed operator for job 1. Fig. 5.9 shows an example of partnership and related schedule bound to an individual in job 1 species (gray color indicates the change time); the machine decision and tool setup in relation to the other 3 job-species. When evolutionary operator is performed on the individual, one operation is selected at random under the probability p_i by following the procedure shown in Fig. 5.7, for example, o_{13} is selected.

The predecessor of o_{13} in Bayesian-network is o_{11}, o_{12} (shown in Fig. 5.9). The marginal probability of o_{12} as well as the conditional probability on machine assignment between o_{12}, o_{13} is presented in Table 5.1 respectively. The probability of o_{13} assigned to machine 4 is $p(x_{134} = 1) = 0.63$, and $p(x_{132} = 1)$ can also be calculated similarly. The machine assignment for o_{13} is determined with respect to $p(x_{132})$ and $p(x_{134})$. If o_{13} is assigned to machine 4, it can be inserted after any operation of the set $\{o_{12}, o_{32}, o_{22}, o_{23}\}$. The position will be inferred from $p\left(z_{ijl} = 1\right) p\left(z_{131} = 1|z_{ijl} = 1\right)$, which consider about the changeover of TAD. When multiple operations are selected, the selected operations will be inferred from the changeover of tool in the same manner as TAD; otherwise, the operations will be selected at random. At this point, the TAD and tool decision are determined respectively. Operation o_{13} is inserted after o_{22}, x_{134}= 1, y_{137} = 1, $z_{13(-x)} = 1$. After iteratively performing the evolutionary operator on the individuals, the individual will be re-evaluated again.

5.3 Experiments and Discussion

Although there has been a significant amount of researches conducted on FJSP (Type 2), there is no benchmarks problem data for the comparison of different algorithms. It is extremely difficult to get real world problem data on FJSP (Type 2). In our study, we use three testing problems namely ZhangNee97, LiMcMahon02 and LiMcMahon07. The specification data of jobs and manufacturing resources are referenced from Zhang et al.[112], Li and McMahon[113], Li and McMahon[114]'s work. The relevant detailed technical specifications of the jobs can be referred from related paper, and it should be noted that the t^{PC}, t^{TC} and t^{DC} of the three problem are set to 140, 20, 120 respectively.

In our experiment, BSEDA will be compared with hrGA, SA, BPSO and BOA that were proposed by Salehi et al. [115], Li et al.[114], Guo et al. [116] and Pelikan [78] to solve the three problems respectively. In order be fair to compare the performance of these algorithms in the same environment, these parameters and strategies related to classification algorithm in Table 5.2 respectively. Furthermore, the same representation is adapted for all of the algorithms, and each algorithm will terminate with the same number of fitness evaluation. All mentioned algorithms were implemented under the Eclipse using the Java programming language and the simulation experiments were analyzed on the environment Intel(R) Core(TM) I7 (3.2 GHz clock) with 4G memory. Meanwhile, in order to avoid concussion, we run each algorithms for 30 times.

Table 5.2 Parameters and strategies of hrGA, SA, BPSO, BSEDA

	hrGA(Salehi et al.[115])	**SA (Li [114])**	**BPSO (Guo et al. [116])**	**BSEDA)**
# of gen	2000	2000	2000	2000
Population	200	-	200	200
Selection	tournament(k)	-	-	tournament(k)
Strategy	elitism	-	elitism	-
Operators	Crossover(P_m) Mutation(P_c)	Shift(P_s) Swapping(P_w) Mutation(P_m)	Shift(P_s) Crossover(P_m) Mutation(P_c)	Bay-based sampling
Parameters	$P_m = 0.65$ P_c= 0.75 $k = 2$	$P_m = 0.65$ $P_c = 0.20$ $P_s = 0.20$	$P_m = 0.65$ $P_c = 0.20$ $P_s = 0.20$ $c_1 = 2, c_2 = 2$	$elitRate = 0.2$ $learningRate = 0.2$ $stThresHold = 20$ $iter = 30$

Table 5.3 Comparison of makespan conducted on hrGA, SA, BPSO, BOA and BSEDA

Problem	**hrGA**			**hrSA**			**BPSO**			**BOA**			**BSEDA**		
	Best	Mean	STD	Best	Mean	STD	Best	Mean	STD	Best	Mean	STD	Best	Mean	STD
ZhangNee97	**1148.0**	1187.2	25.5	**1148.0**	1188.2	25.4	**1148.0**	1182	22.2	**1148.0**	1159.8	12.2	**1148.0**	1162.8	15.4
LiMcmahon0	**1128.0**	1155.0	26.2	**1128.0**	1156.5	25.9	**1128.0**	1146.8	20.1	**1128.0**	1133.0	8.4	**1128.0**	1139.6	11.6
LiMcmahon0	1309.0	1633.3	188.9	1376.0	1754.8	209.8	1210.0	1433.6	173.2	**1106.0**	1179.6	20.7	1132.0	1221.8	45.1

To investigate the performance of BSEDA for different facets of problem difficulties, experiments are divided into five parts based on performance analyzing objectives.

(1) Computation Efficiency and Optimality on benchmark LiMcMahon07 : Evaluating the computation times and convergence speed of the proposed BSEDA, BOA and BPSO respectively.

(2) Computation Efficiency on Benchmark set 3 benchmark: Evaluating the computation times of the proposed MREDA and BOA respectively.

(3) Optimality on Benchmark set 3 benchmark sets: Evaluating the optimality of the proposed MREDA and BOA respectively.

(4) Optimality Stability Analysis: Evaluating the dispersion of the proposed with BSEDA , hrSA, hrGA, BPSO and BOA respectively.

(5) Parameter tuning: The parameters used in MREDA are roughly divided into three types: learning MR, MR-based sampling and EDA control parameter such as truncation selection as promising solutions. In this paper, the maximum number of *k*-neighbors will be investigated.

5.3.1 Computation Efficiency and Optimality on benchmark LiMcMahon07

To evaluate computation efficiency and optimality on one benchmark in detail. We evaluate The convergence conducted on LiMcmahon07 is show in Fig. 5.10. Compared with BOA, Bayesian network is constructed by using a chosen metric and constraints in each generation in BOA. BOA employs greedy strategy-based learning algorithm. With the size of the degree of freedom on flexibility being larger, the solving time of BOA is too long and difficult to apply to the practical problem within the scenario considered. For convergence, it presents that the proposal BSEDA reduces computation costs (185%) than BOA under reaching the same value of VTR (1230). On the other hand, Our proposal achieves near-optimal (worse3.58%) than BOA when iteration reaches 2000.

Moreover, For convergence, it presents that the proposal BSEDA reduces computation costs (28%) than BPSO under reaching the same value of VTR (1230). On the other hand, Our proposal achieves better optimality (improved 19%) than BPSO when iteration reaches 2000. While BSEDA takes more computation costs because of learning the Bayesian network. The snapshot of experiment is shown in Fig. 5.12.

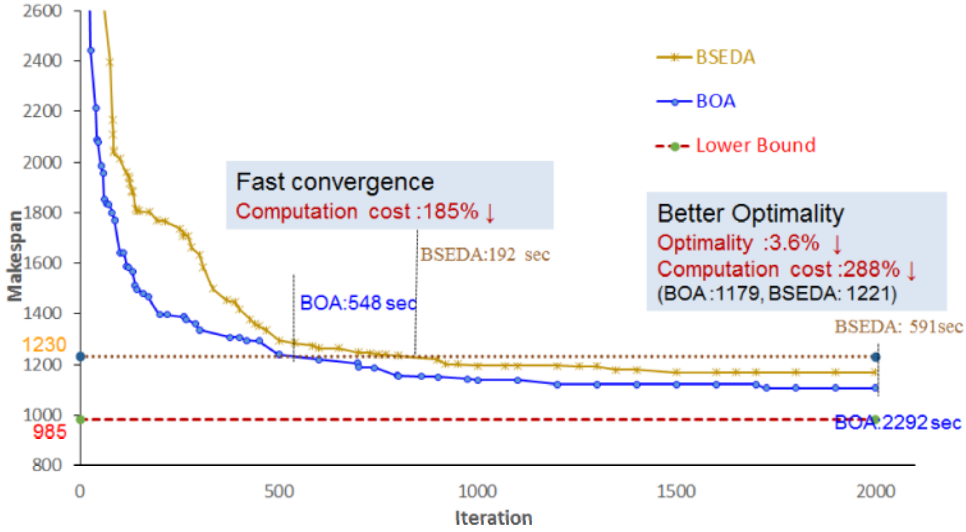


Fig. 5.10 Convergence of BSEDA and BOA conducted on LiMcmahon07

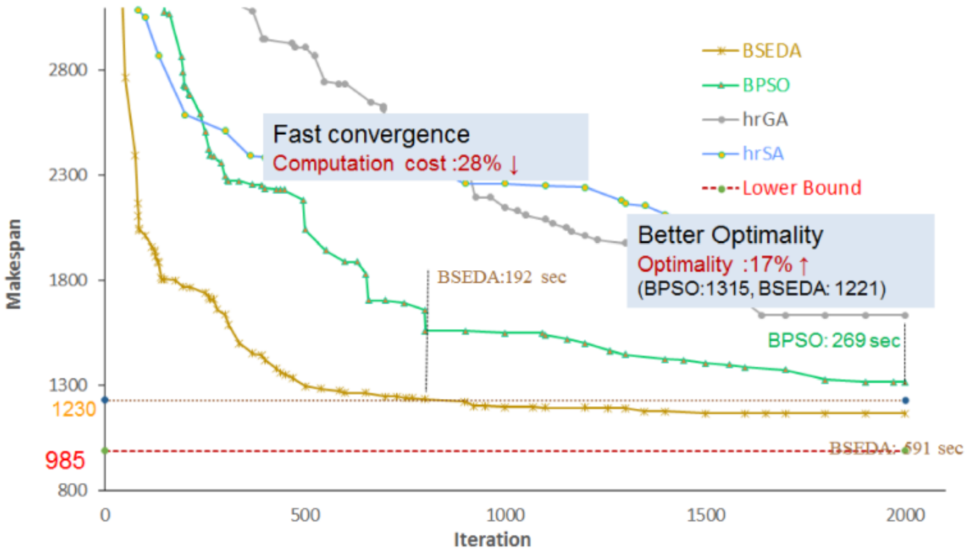


Fig. 5.11 Convergence of BSEDA, hrGA, hrSA and BPSO conducted on LiMcmahon07

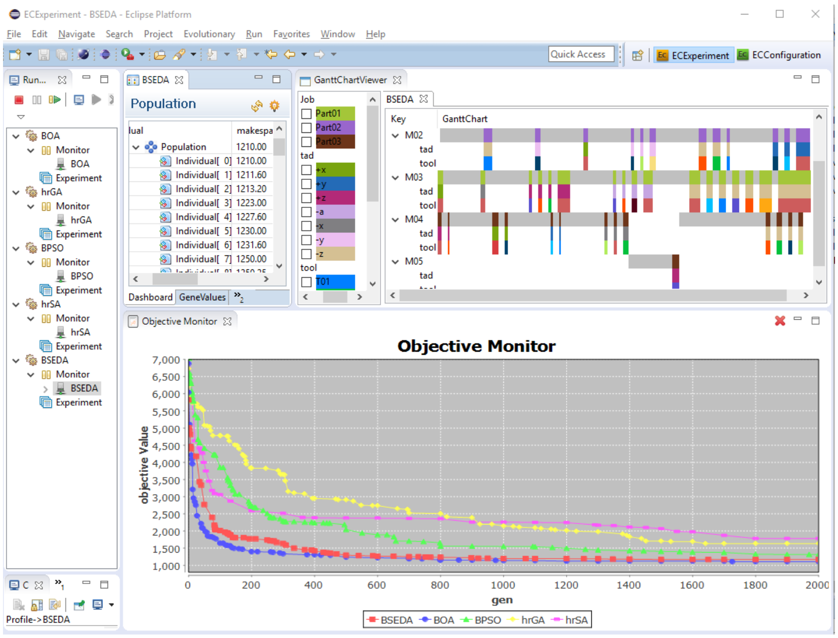


Fig. 5.12 Snapshot of experiments conducted on LiMcmahon07

It should be noted that, In BSEDA, Bayesian network is constructed by using a chosen metric and constraints in each generation in BSEDA. With the size of the degree of freedom on flexibility being larger, the solving time of BSEDA is too long and difficult to apply to the practical problem within the scenario considered.

5.3.2 Computation Efficiency

The calculating cost based on evolutionary algorithm mainly depends on the number of health assessment. There exists the differences among Salehi, Li, Guo, Pelikan and our method on time complexity, depends on the knowledge-base operators. In particularly, BSEDA involves Bayesian-based operator and needs to train the Bayesian model using the promising solutions. It depends largely on X_{ij}, Y_{ij}, and Z_{ij} which represents Machine Flexibility (MF), Tool Flexibility (TF) and TADs Flexibility (AF) respectively. The calculation cost of training can roughly estimate by the Bayesian model equation:

$$\begin{aligned}&\textbf{COST} \propto \\ &|num\ of\ gen| \times |num\ of\ popSize| \times \\ &|numofoperations| \times \\ &|learning\ frequency| \times \\ &|degree\text{-}of\text{-}freedom\ on\ MF, TF, AF|\end{aligned}$$

Table 5.4 Comparison on Computation Cost (Sec)

Problem	hrGA	hrSA	BPSO	BOA	BSEDA
ZhangNee97	84.47	109.36	75.25	107.46	65.12
LiMcmahon02	127.20	145.20	114.69	145.52	104.71
LiMcmahon07	307.84	291.19	269.34	548.91	192.58

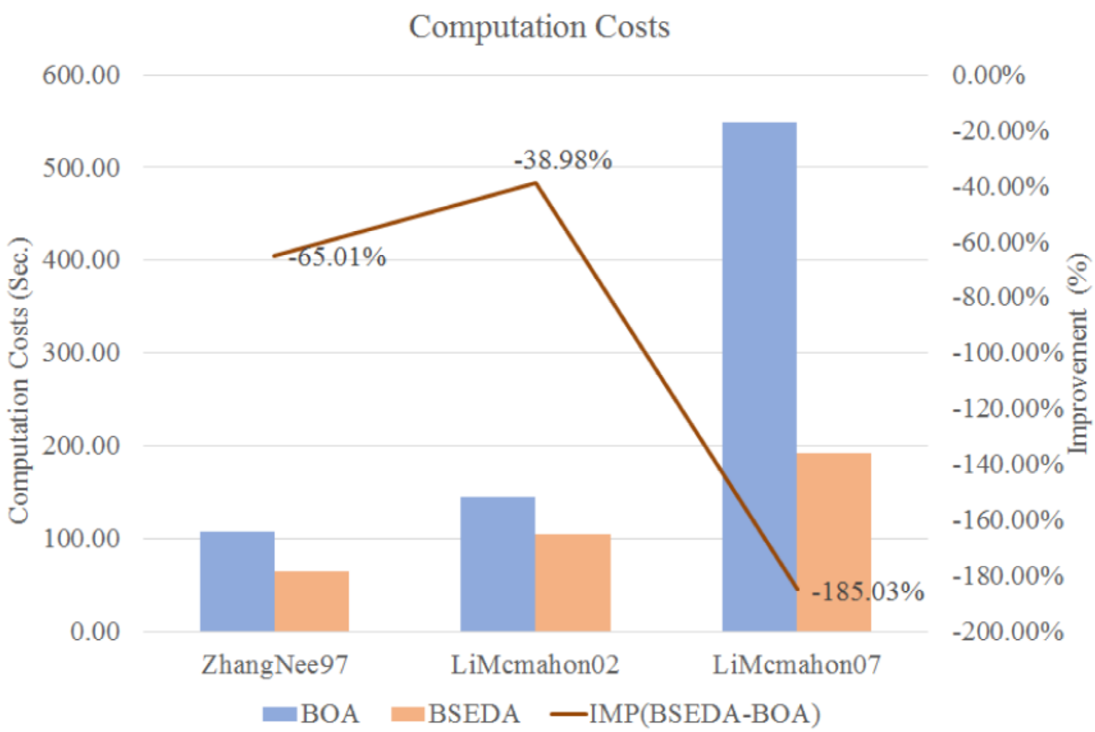


Fig. 5.13 Comparison of average CPU Time between the proposal and other approaches

In practice, the degree of freedom on the MF and AF is not always more than 10. The computation cost is reasonably acceptable with respect to the construction of Bayesian network as given in FJSP (Type 2) (shown in Table 5.4). It should be noted that the value of VTR for ZhangNee97, LiMcmahon02, LiMcmahon07 is 1189, 1159 and 1230 respectively. Table 5.3 shows that BSEDA achieves better solutions than SA and BPSO and BSEDA, Within considerable computation time, its convergence speed outperforms hrGA, SA, and BPSO. This claim is supported by the experiment conducted on the LiMcMahon07 example and that subsequent results are shown in Fig. 5.13. Fig. 5.13 shows that BSEDA outperforms the other approaches on all test problems in terms of computation costs. Particularly, comparison to BOA, the best meta-heuristic solver even, the computation costs of the proposal is on average 185% faster than BOA.

5.3.3 Optimality

Table 5.3 shows that BSEDA performed better than hrGA, hrSA, and BPSO respectively in all the experiments. hrSA is outperformed by BSEDA as it adopts a slower cooling and more iterations strategy in Li's approach. By manipulating the cooling schedule, SA can lead to better final solution and possess a form that is convergence proof. However, the trial number of performing neighborhood operator significantly affects the SA's performance. Moreover, the hrSA iterates a single point using a unary neighborhood operator, for large scale problem, the SA-based algorithm shows lack of ability in exploration when compared with multiple points based algorithm such as hrGA and BPSO (shown in Fig. 5.15c).

On the other hand, BPSO-based algorithm can achieve better results than traditional GA-based algorithms. GA iterates an entirely population just by using two-parent operator crossover as well as one-parent operator mutation, while BPSO-based approach shares the local best and the global best decision value on each dimension of the solution, so traditional EAs is inherently biased against excessive exploitation, and lose robustness and stability in a statistical sense (shown in Fig. 5.15).

Unfortunately, these approaches discussed here seldom consider evolutionary operators on the original solution, and limited actions are taken to reduce the side effects caused by sequence-dependent. BSEDA offer the variant of decision variables a forecasting method about machine assignment and tool setup on the machine for FJSP (Type 2), and they achieve better stability than random strategy based algorithms (shown in Fig. 5.15), although the effects of the accuracy of the prediction is a promising solution.

5.3.4 Stability Analysis

To evaluate the dispersion of BSEDA, we compare BSEDA, hrGA, hrSA, BPSO and BOA conducted on ZhangNee97, LiMCMahon02 and LiMCMahon07. The dispersion performances, as shown in Fig. 4.15 indicates that BSEDA is obviously better than hrGA, hrSA and BPSO, and BSEDA can also achieve satisfactory dispersion performance.

BSEDA and BOA provide a prediction mechanism on the variant of decision variables about machine assignment and tool setup on the machine for FJSP (Type 2), and they achieve better stability than random strategy based algorithms (shown in Fig. 5.15), although the accuracy of prediction is affected by the promising solutions. The comparison of optimality and improvement on the make-span is show in Fig. 5.14.

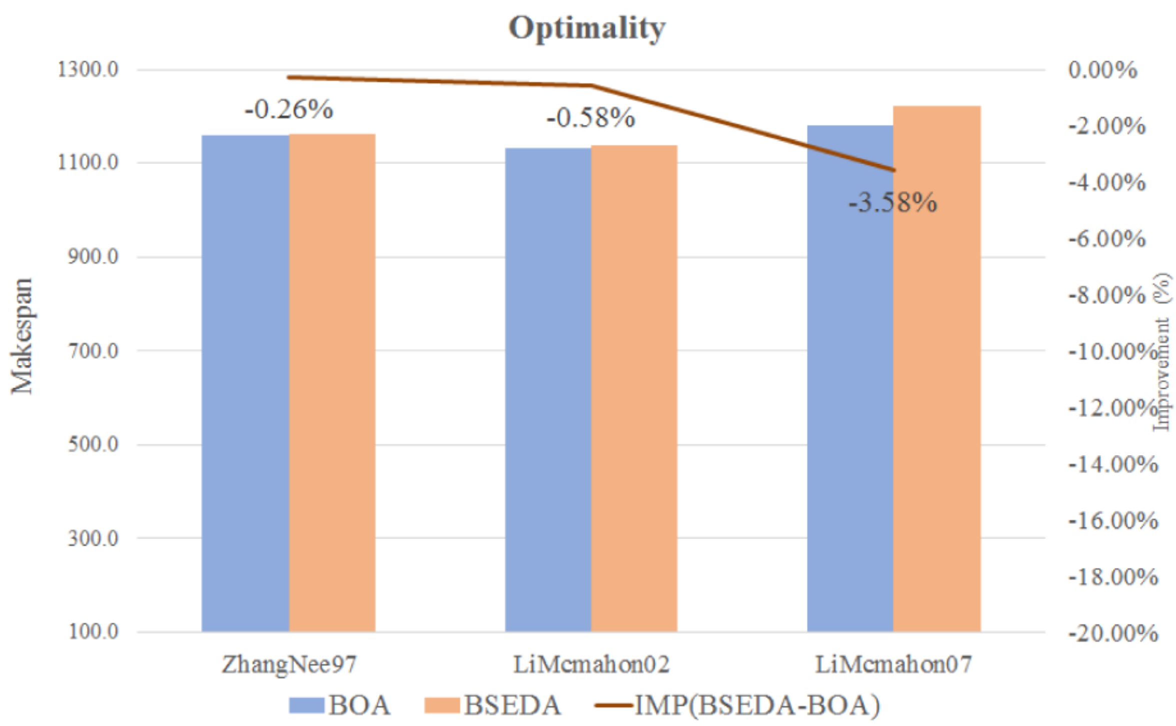


Fig. 5.14 Comparison of optimality between the proposal and other approaches

Table 5.5 Comparison of BSEDA (withLocal) and BSEDA (withoutLocal) on Makespan

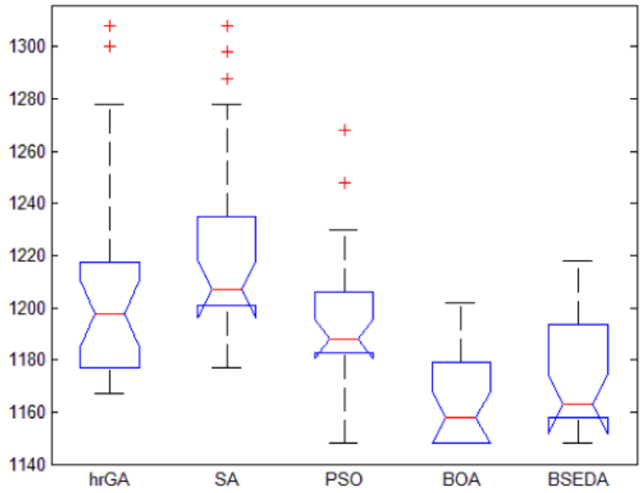
Problem	BSEDA (withLocal)			BSEDA (withoutLocal)		
	Best	Mean	STD	Best	Mean	STD
ZhangNee97	**1148.0**	1162.8	15.4	**1148.0**	1198.8	19.5
LiMcmahon02	**1128.0**	1139.6	11.6	**1128.0**	1178.6	24.1
LiMcmahon07	**1132.0**	1118.0	45.1	1132.0	1271.0	78.7

Effectiveness of local search

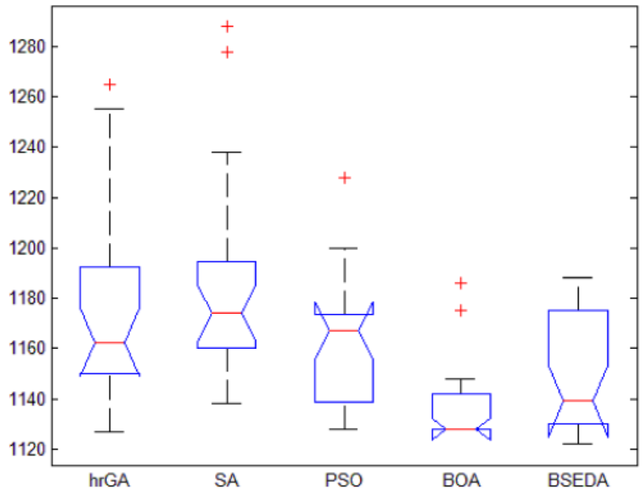
In order to analyze influence of the GA-based local search during the search of BSEDA, We do experiments both with local search algorithm and without under the previous problems for both of algorithms. The record mean and variance value of makespan marked the outstanding performance in bold, as shown in Table 5.5 and Fig. 5.13 . Table 5.5 shows that the result of BSEDA (withoutLocal) outperforms BSEDA, and outperforms BPSO for three benchmarks.

5.3.5 Parameter Tuning

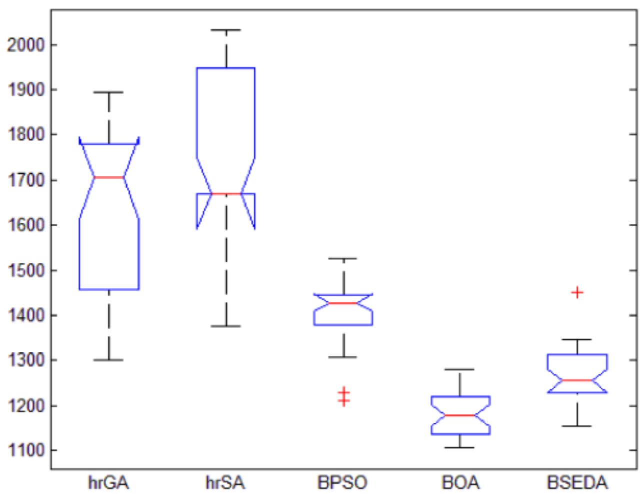
There exists three groups of parameters of roughly: learning Bayesian-network, Bayesian-based operator, and GA-based local optimization. In BSEDA, Bayesian network plays a



(a) ZhangNee97



(b) LiMcMahon02



(c) LiMcMahon07

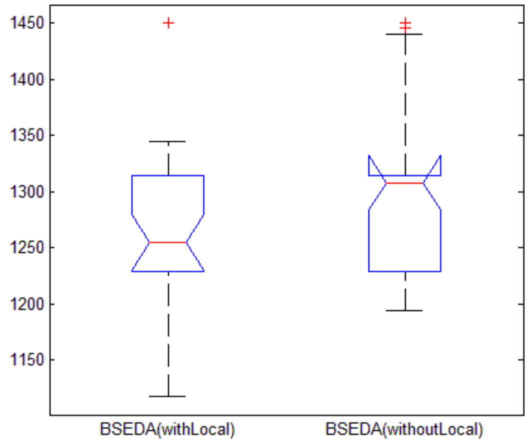


Fig. 5.16 Boxplot of Makespan by BSEDA (withLocal) and BSEDA (withoutLocal)

critical role to estimate a probability distribution in order to generate a set of sample solution in the next generation. The learning rate α affects the portions of the search space to be explored and also affects the speed at which the probability distribution is shifted to the promising data.In consideration of the length of the paper, discussion of the test results analysis only focused on the learning rate Settings, and other parameters share the same configuration as shown in Table 5.2. It is shown that that α changes from 0.1 to 0.20 in Fig. 5.17, BSEDA can get better performance. If α is too large, it may lead to premature convergence. On the contrary, while α is too small, it may cost more time to find the optimal solution (shown in Fig. 5.18). It should be noted that BSEDA evolved into co-evolutionary mechanism with random seeking strategy when $\alpha = 0$, and it got better results than traditional EA algorithms.

5.4 Summary

In this case study, we study BSEDA for overcoming the challenges of modeling and evaluating the complexity of FJSP (Type 2). FJSP (Type 2) considers manufacturing environment with multi-purpose machines equipped with tool-box, and the tool changeover and job changeover have highly depended on job processing sequence. Bayesian network is employed to learn machine allocations of the operations affecting the make-span, simultaneously model the parenthood relationship between tools equipped with the machine. A positive result shows

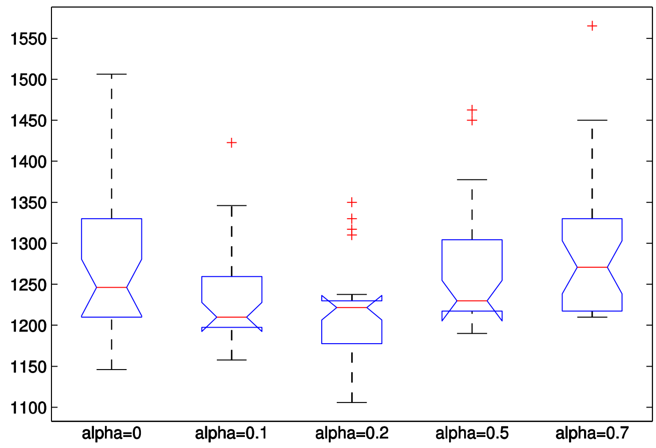


Fig. 5.17 Variation of Makespan tunning by learning rates

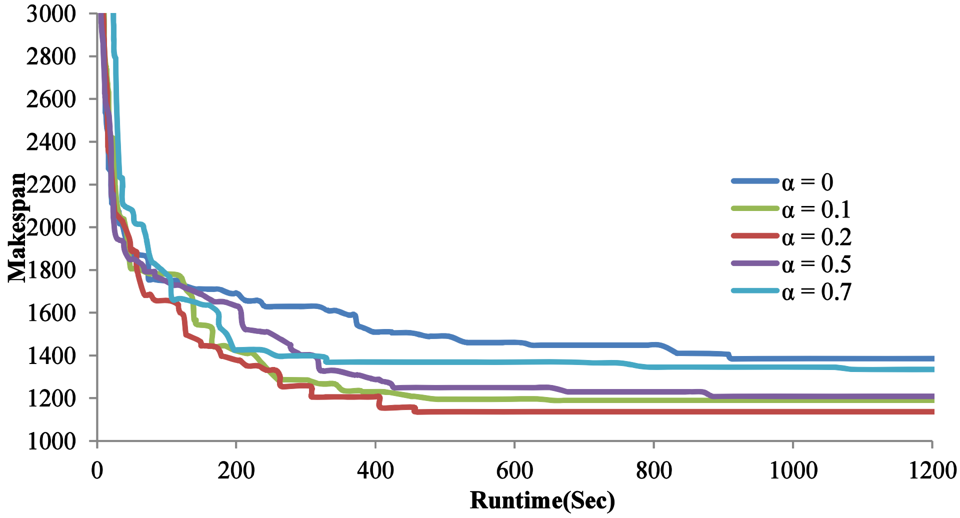


Fig. 5.18 Convergences trend of tunning on learning rates

that BSEDA method is applicable to a dynamic operating environment to provide decision support.

Chapter 6

Conclusions

6.1 Conclusion and summary

In a world of increasing global competitiveness, the current manufacturing must overcome challenges associated with the middle size batch process consisting of a variety of products, and short product life cycles, particularly for electronic industry. In this situation, improvement of operational processes is likely to major drivers to realize the necessary cost reduction and meet the reduced cycle time. Consequently, the scheduling dealing with product-mix processing and job transportation is ultimately required for the agile manufacturing system (AMS).

Scheduling problem is a classical one of NP-hard discrete optimization problems. In last decades, intelligent manufacturing planning and scheduling based on meta-heuristics, such as Genetic Algorithms (GAs), Simulated Annealing (SA) , Binary Particle Swarm Optimization (BPSO), have become common tools for finding satisfactory solutions within reasonable computational times in real settings. Recently, Discrete Estimation of Distribution Algorithms (EDA), a probabilistic model-based stochastic optimization method, was proposed to solve the scheduling problems and achieve better performance. However, these approaches assume that the decision variable is independent, and the optimality has significantly decreased for the problem with interdependence among the dimensional variables. On the other hand, research work steadily grows to learn the relationship and structure between the variables by machine learning based on the data.

In this dissertation, incorporating machine learning technique into metaheuristics for combinatorial problems optimization, we propose a novel meta-heuristic algorithm of hybrid EDA based on Probabilistic Graphical Models (GPM) which is employed to find the interdependence among decision variables by using machine learning techniques. Based on two different GPMs, two types of hybrid EDA are proposed in this dissertation. One is Markov random field-based EDA (MREDA) for solving scheduling problems which include combination interdependence relationship among decision variables, and the other one is Bayesian network-based EDA (BSEDA) for solving scheduling problems which include the sequential dependence relationship among decision variables. To confirm the effectiveness of our approaches, we applied the proposed method to solve well-studied scheduling problems: flexible job shop scheduling problem where the setup time of operation is negligible (FJSP Type 1), and flexible job shop scheduling problem where the setup time of operation is not negligible (FJSP Type 2), and we proved the effectiveness of the proposed methods by comparative experiments with conventional meta-heuristic scheduling methods.

In chapter 2 [Literature review], firstly, gives an overview of EDAs and the application related to scheduling problems. Particularly, we survey the previous approaches according to the probabilistic model. Next, we briefly introduce the literature of Probabilistic Graphical Model (Markov Random Field and Bayesian network) concerning machine learning techniques. Bayesian network can be used for modeling the cause and effect relationship between the decision variables. Markov network can be used to model interdependency of decision variables.

In chapter 3 [problem description and approach for solving problems], for the scheduling problem of the manufacturing environment consists of multiple-functional machines providing flexibility for processing the jobs. Inherently, there are two types of decision interdependencies in resource allocation problems: (1) resource allocation to combination-dependency (2) sequence-dependency. However, this kind of many-to-many decision interdependencies in resource allocation are not efficiently considered in traditional EDAs. Therefore, we propose a novel EDA based on network- graphical model to construct those decision interdependencies in resource allocation problems.

Firstly, we proposed Markov random field-based EDA (MREDA). In MREDA, Markov network is used to model combination-dependency of decisions concerning resource allocation of operations. MREDA takes care of exploration that tries to identify the most promising search space regions, and to model conditional dependence of the resource allocation variables. The conditional probabilities defined by the local Markov property are estimated, and the new candidate solutions are sampled according to the given sampling method. For a candidate solution, problem specific based local search algorithm is used to improve each candidate solution to reach a local optimum.

Secondly, we propose Bayesian network based EDA (BSEDA). Different to the traditional chain, tree model, Bayesian network in BSEDA is employed to model the many-to-many sequence-dependencies of resource allocation. It can be more efficient to exploit resource allocation considering changoever along the job process routing. To enhance the ability to solve large-scale problems of high-dimensional optimization; Especially for local exploitation, thus, we consider using two sub-populations based GA-based algorithm to adjust the machine assignment and operation sequence respectively with a splitting criterion and a combination criterion.

In chapter 4 [Markov random field-based EDA (MREDA) and its scheduling application], to evaluate of the merit of proposed MREDA, MREDA is employed to solve flexible job-shop scheduling problem (FJSP) with minimizing make-span. FJSP is expanded from the traditional JSP. Machine allocation to per operation in FJSP can be formulated as a many-to-many network model. Thus, Markov property described by an undirected graph can be efficiently applied to FJSP. The proposed MREDA method compared with the latest algorithm BEDA proposed by Wang et al. In Wang' research work, it has been shown that BEDA outperforms published conventional Algorithms such as KBACO, BPSO and hybrid tabu search, etc. Numerical simulations are carried out with widely used benchmark 10 BRdata instances, 18 DPdata instances and 21 BCdata instances. For optimality, MREDA outperforms BEDA on all of the test cases. The optimality concerning makespan is efficiently improved about 4% for middle and large-scale problem. For computation efficiency, the average computation costs of the proposal is about 15% faster than BEDA. Moreover, MREDA presents more stable than BEDA.

In chapter 5 [Bayesian network-based EDA (BSEDA) and its scheduling application], to evaluate the proposed BSEDA, Experiment is conducted to solve flexible job shop scheduling problem where the setup time of operation is not negligible (FJSP Type 2) with minimizing make-span. Different with FJSP(Type 1), FJSP (Type 2) considers manufacturing environment with multi-purpose machines equipped with tool-box, and the tool changeover and part changeover have highly depended on job processing sequence. Bayesian network is employed to learn machine allocations of the operations affecting the make-span, simultaneously model the parenthood relationship between tools equipped with the machine. The proposed BSEDA compared with recently published conventional algorithms: hrGA, hrSA, BPSO and BOA. We use three testing problems namely ZhangNee97, LiMcMahon02 and LiMcMahon07. For optimality, BSEDA performed better than hrGA, SA, BPSO, BOA in all the experiments, the optimality concerning makespan is improved about 17% than BEDA for large-scale problem. For computation efficiency, the computation costs of the proposal is 28% faster than the best previous approach BPSO. Moreover, comparison to greedy-strategy-based BOA, BSEDA can reduce 185% computation time than BOA while achieving same near-optimal solution.

6.2 Future work

From the previous investigation on hybridized PGM-based EDA optimization approaches to resource-constraint scheduling problems in manufacturing system. Although these approaches achieve some attractive solutions and easily apply to other types of scheduling problems such as project scheduling and nurse scheduling, there is still much space to improve. In the present work, we just consider single objective optimization. More problem-dependent considerations are concerned and one global optimal solution and less computation time are desired. However, *multiobjective optimization problem* (moOP) needs additional cares in *convergence* mechanism of finding sufficient number of global Pareto optimal solutions as soon as possible, and *dispersion* mechanism of distributing them as evenly as possible. Moreover, reduction of *computation time* becomes more difficult accordingly. For future work, we would like to conduct multiple objectives optimization to extend thesis work.

On the other hand, The accuracy of estimated PGM model has the great affect on the performance of EDA. Therefore, further research will improve the performance of both MREDA and BSEDA, particularly for the implementation of more efficient structure learning algorithms, and also for the use of other more efficient variations of the Gibbs sampler algorithm. Although the approaches adopts an integrated approach for solving manufacturing scheduling problem, it is crucial to analyze the effects of interdependent relationships existing between the machine assignments and the operation sequence.

Acknowledgements

First and foremost, I would like to express my heartfelt gratitude to my advisor, Prof. Tomohiro MURATA, for his great guidance and tireless support over the past few years. I am greatly thankfull that he would take me as a student at the difficult period of my PhD. I learned the skills of conducting academic research, writing a mathematical paper and presenting my work to others effectively and clearly from the acadamic activities. These skills have the benefit of my career at present and future. I am greatly indebted to Professor Hee-Hyol LEE of Waseda University and Professor Shigeru FUJIMURA of Waseda University, the committee members, for their careful reviews, constructive advices and helpful suggestions in my work.

I would like to show my deepest appreciation to all my advisor at Waseda University for for their invaluable assistance and patience. In particular, I would like to thank Howie LIN, Xili CHEN, Jing TIAN and Shuang WU. We have spent many long nights together, reading each other's paper drafts and discussing the ideas of the related research topices. Working with each one of them has been an unforgettable learning experience. I gratefully acknowledge my dear friends Lin LIN and Wenqiang ZHANG whose meticulous suggestions and comments were of inestimable value to this study.

I would also like to thanks to my friends and my families for all their love and support. My friends Jinghong DING, Ke WANG, ShiTao YAN, Gaofeng ZHENG, Cong WANG, Lu SUN and Yan WANG deserve my gratitude for their selfless support. In particular, I owe many thanks to Yongmei WANG for taking care of my family in hometown. Last but not least, my special thanks goes to my wife, Yanjie CHEN and everyone else in my family: my parents, my daughter, Qiqi and my son, Chenchen. They have supported me during PhD by means that words cannot be explained.

Publications

Journals

[1] Tian Jing, Xinchang Hao, Tomohiro Murata. "Markov Network based Multi-objective EDA and its Application for Resource Constrained Project Scheduling", IEEJ Transactions on Electronics, Information and Systems, Vol.136, No.3, pp. 290–298 , 2016.

[2] Xinchang Hao, Tian Jing, Hao Wen Lin, Tomohiro Murata,"An Effective Markov Random Fields based Estimation of Distribution Algorithm and Scheduling of Flexible Job Shop Problem", IEEJ Transactions on Electronics, Information and Systems, Vol.134, No.6, pp. 796–805, 2014.

[3] Xinchang Hao, Xili Chen, Hao Wen Lin, Tomohiro Murata, "Cooperative Bayesian Optimization Algorithm: a Novel Approach to Multiple Resources Scheduling Problem", IEEJ Transactions on Electronics, Information and Systems, Vol.132, No.12, pp.2007-2018, 2012.

[4] Xili Chen, Xinchang Hao, Hao Wen Lin, Tomohiro Murata, "Multi objective dynamic job shop scheduling using composite dispatching rule and reinforcement learning", IEEJ Transactions on Electronics, Information and Systems, Vol.131, No.6, pp.1241-1249, 2011.

[5] Xinchang Hao, M. Gen, "Multi-objective Job Shop Rescheduling with Evolutionary Algorithm", IEEJ Transactions on Electronics, Information and Systems, Vol.131, No.3, pp.674–681, 2011.

International Conference

[1] Jing Tian, Xinchang Hao, Tomohiro Murata, "Multi-objective EDA for Multi-mode Project Scheduling with Resource Constraint" in Asian Conference on Information Systems (ACIS), 2015. The Second International Conference, pp. 7–12, Nov. 2015.

[2] Jing Tian, Xinchang Hao, Tomohiro Murata, “Markov Network based EDA and Robust Project Scheduling with Multi-mode Resource Constraint and Rework Uncertainty”, in Asian Conference on Information Systems (ACIS), 2014. The Third International Conference, pp.363-370, 2014.

[3] Xinchang Hao, Jing Tian, Hao Wen Lin, Tomohiro Murata, “Markov Random Fields based Estimation of Distribution Algorithm and Application of Sharable Resource Constrained Scheduling Problem” in Asian Conference on Information Systems(ACIS), 2013. The Second International Conference, pp. 62–68, Nov. 2013.

[4] Jing Tian, Xinchang Hao, Tomohiro Murata, “Robust Scheduling by Hybrid Method of Policy Tree and Estimation of Distribution Algorithm and Application to Resource Constrained Project Scheduling Problem with Rework Uncertainty” in Asian Conference on Information Systems(ACIS), 2013. The Second International Conference, pp. 371–378, Nov. 2013.

[5] Xinchang Hao, Hao Wen Lin, Jing Tian, Tomohiro Murata, “Cooperative Estimation of Distribution Algorithm for Flexible Job shop Problem with Configurable Machines,” in Asian Conference on Information Systems(ACIS), 2012. The First International Conference, pp. 239 –244, Dec. 2012.

[6] Xinchang Hao, Xili Chen, Hao Wen Lin, Tomohiro Murata, “Cooperative Bayesian Optimization Algorithm: A Novel Approach to Simultaneous Multiple Resources Scheduling Problem,” in Innovations in Bio-inspired Computing and Applications (IBICA), 2011. The Second International Conference, pp. 212 –217, Dec. 2011.

[7] Xinchang Hao, Hao Wen Lin, and Tomohiro Murata, “Application of Negotiable Evolutionary Algorithm in flexible manufacturing planning and scheduling,” in Industrial Engineering and Engineering Management (IE&EM), 2010. IEEE 17Th International Conference, pp. 496–500, Nov. 2010.

[8] Xili Chen, Xinchang Hao, Hao Wen Lin, Tomohiro Murata, “Rule driven multi objective dynamic scheduling by data envelopment analysis and reinforcement learning”, Proceeding of ICAL2010(2010 IEEE International Conference on Automation and Logistics), pp.396-401, Aug. 2010.

References

[1] Mikell P Groover. *Automation, production systems, and computer-integrated manufacturing*. Prentice Hall Press, 2007.

[2] Mark Hauschild and Martin Pelikan. An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, 1(3):111–128, 2011.

[3] Pedro Larranaga and Jose A Lozano. *Estimation of distribution algorithms: A new tool for evolutionary computation*, volume 2. Springer Science & Business Media, 2002.

[4] Ryszard S Michalski, Jaime G Carbonell, and Tom M Mitchell. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

[5] YW Guo, WD Li, Antony R Mileham, and Geraint W Owen. Applications of particle swarm optimisation in integrated process planning and scheduling. *Robotics and Computer-Integrated Manufacturing*, 25(2):280–288, 2009.

[6] Melanie Mitchell, Stephanie Forrest, and John H Holland. The royal road for genetic algorithms: Fitness landscapes and ga performance. In *Proceedings of the first european conference on artificial life*, pages 245–254. Cambridge: The MIT Press, 1992.

[7] David E Goldberg and John H Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[8] Lawrence J Fogel, Alvin J Owens, and Michael J Walsh. Intelligent decision making through a simulation of evolution. *Behavioral science*, 11(4):253–272, 1966.

[9] Hans-Paul Paul Schwefel. *Evolution and optimum seeking: the sixth generation*. John Wiley & Sons, Inc., 1993.

[10] John R Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

[11] John R Koza. Genetic programming ii: Automatic discovery of reusable subprograms. *Cambridge, MA, USA*, 1994.

[12] James Kennedy. Particle swarm optimization. In *Encyclopedia of Machine Learning*, pages 760–766. Springer, 2010.

[13] James Kennedy, James F Kennedy, Russell C Eberhart, and Yuhui Shi. *Swarm intelligence*. Morgan Kaufmann, 2001.

[14] Marco Dorigo and Christian Blum. Ant colony optimization theory: A survey. *Theoretical computer science*, 344(2):243–278, 2005.

[15] John J Grefenstette. Optimization of control parameters for genetic algorithms. *Systems, Man and Cybernetics, IEEE Transactions on*, 16(1):122–128, 1986.

[16] Heinz Mühlenbein and Gerhard Paass. From recombination of genes to the estimation of distributions i. binary parameters. In *Parallel Problem Solving from Nature—PPSN IV*, pages 178–187. Springer, 1996.

[17] Zbigniew Michalewicz. *Genetic algorithms+ data structures= evolution programs*. Springer Science & Business Media, 2013.

[18] Jeremy S De Bonet, Charles L Isbell, Paul Viola, et al. Mimic: Finding optima by estimating probability densities. *Advances in neural information processing systems*, pages 424–430, 1997.

[19] Shumeet Baluja and Scott Davies. Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. Technical report, DTIC Document, 1997.

[20] CK Chow and CN Liu. Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on*, 14(3):462–467, 1968.

[21] Martin Pelikan and Heinz Mühlenbein. The bivariate marginal distribution algorithm. In *Advances in Soft Computing*, pages 521–535. Springer, 1999.

[22] Jean Dickinson Gibbons and Subhabrata Chakraborti. *Nonparametric statistical inference*. Springer, 2011.

[23] Georges Harik. Linkage learning via probabilistic modeling in the ecga. *Urbana*, 51(61):801, 1999.

[24] Martin Pelikan, David E Goldberg, and Erick Cantu-Paz. Linkage problem, distribution estimation, and bayesian networks. *Evolutionary computation*, 8(3):311–340, 2000.

[25] Zukui Li and Marianthi Ierapetritou. Process scheduling under uncertainty: Review and challenges. *Computers & Chemical Engineering*, 32(4):715–727, 2008.

[26] Yong Wang, Jian Xiang, and Zixing Cai. A regularity model-based multiobjective estimation of distribution algorithm with reducing redundant cluster operator. *Applied Soft Computing*, 12(11):3526–3538, 2012.

[27] Peter B Luh, Debra J Omt, Eric Max, and Krishna R Pattipati. Schedule generation and reconfiguration for parallel machines. *Robotics and Automation, IEEE Transactions on*, 6(6):687–696, 1990.

[28] Peter B Luh and Debra J Hoitomt. Scheduling of manufacturing systems using the lagrangian relaxation technique. *Automatic Control, IEEE Transactions on*, 38(7):1066–1079, 1993.

[29] Debra J Hoitomt, Peter B Luh, and Krishna R Pattipati. A practical approach to job-shop scheduling problems. *Robotics and Automation, IEEE Transactions on*, 9(1):1–13, 1993.

[30] Haoxun Chen, Chengbin Chu, and Jean-Marie Proth. An improvement of the lagrangean relaxation approach for job shop scheduling: a dynamic programming method. *Robotics and Automation, IEEE Transactions on*, 14(5):786–795, 1998.

[31] Haoxun Chen and Peter B Luh. An alternative framework to lagrangian relaxation approach for job shop scheduling. *European Journal of Operational Research*, 149(3):499–512, 2003.

[32] Peter B Luh, Xing Zhao, Yajun Wang, and Lakshman S Thakur. Lagrangian relaxation neural networks for job shop scheduling. *Robotics and Automation, IEEE Transactions on*, 16(1):78–88, 2000.

[33] Shrikant S Panwalkar and Wafik Iskander. A survey of scheduling rules. *Operations research*, 25(1):45–61, 1977.

[34] Mitsuo Gen, Runwei Cheng, and Lin Lin. *Network models and optimization: Multiobjective genetic algorithm approach*. Springer Science & Business Media, 2008.

[35] Nhu Binh Ho, Joc Cing Tay, and Edmund M-K Lai. An effective architecture for learning and evolving flexible job-shop schedules. *European Journal of Operational Research*, 179(2):316–333, 2007.

[36] Joc Cing Tay and Nhu Binh Ho. Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. *Computers & Industrial Engineering*, 54(3):453–473, 2008.

[37] Joseph Adams, Egon Balas, and Daniel Zawack. The shifting bottleneck procedure for job shop scheduling. *Management science*, 34(3):391–401, 1988.

[38] Lars Mönch, Rene Schabacker, Detlef Pabst, and John W Fowler. Genetic algorithm-based subproblem solution procedures for a modified shifting bottleneck heuristic for complex job shops. *European Journal of Operational Research*, 177(3):2100–2118, 2007.

[39] Jie Gao, Mitsuo Gen, Linyan Sun, and Xiaohui Zhao. A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems. *Computers & Industrial Engineering*, 53(1):149–162, 2007.

[40] Hsueh-Chien Cheng, Tsung-Che Chiang, and Li-Chen Fu. A two-stage hybrid memetic algorithm for multiobjective job shop scheduling. *Expert systems with applications*, 38(9):10983–10998, 2011.

[41] Lester Ingber. Simulated annealing: Practice versus theory. *Mathematical and computer modelling*, 18(11):29–57, 1993.

[42] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[43] Christos Koulamas, SR Antony, and R Jaen. A survey of simulated annealing applications to operations research problems. *Omega*, 22(1):41–56, 1994.

[44] Taicir Loukil, Jacques Teghem, and Philippe Fortemps. A multi-objective production scheduling case study solved by simulated annealing. *European journal of operational research*, 179(3):709–722, 2007.

[45] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5):533–549, 1986.

[46] Pierre Hansen. The steepest ascent mildest descent heuristic for combinatorial programming. In *Congress on numerical methods in combinatorial optimization, Capri, Italy*, pages 70–145, 1986.

[47] Wojciech Bożejko, Mariusz Uchroński, and Mieczysław Wodecki. The new golf neighborhood for the exible job shop problem. *Procedia Computer Science*, 1(1):289–296, 2010.

[48] Jun-qing Li, Quan-ke Pan, and Yun-Chia Liang. An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering*, 59(4):647–662, 2010.

[49] Qiao Zhang, Hervé Manier, and M-A Manier. A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times. *Computers & Operations Research*, 39(7):1713–1723, 2012.

[50] Mark S Fox and Stephen F Smith. Isis—a knowledge-based system for factory scheduling. *Expert systems*, 1(1):25–49, 1984.

[51] Claude Lepape. Soja: a daily workshop scheduling system, soja's system and inference engine. In *Proc. of the fifth technical conference of the British Computer Society Specialist Group on Expert Systems on Expert systems 85*, pages 195–211. Cambridge University Press, 1986.

[52] Stephen F Smith, Mark S Fox, and Peng Si Ow. Constructing and maintaining detailed production plans: Investigations into the development of kb factory scheduling. *AI magazine*, 7(4):45, 1986.

[53] Mark S Fox and Norman M Sadeh. Why is scheduling difficult? a csp perspective. In *ECAI*, pages 754–767, 1990.

[54] Weiming Shen, Qi Hao, Hyun Joong Yoon, and Douglas H Norrie. Applications of agent-based systems in intelligent manufacturing: An updated review. *Advanced engineering INFORMATICS*, 20(4):415–431, 2006.

[55] Lawrence Davis. Job shop scheduling with genetic algorithms. In *Proceedings of an international conference on genetic algorithms and their applications*, volume 140. Carnegie-Mellon University Pittsburgh, PA, 1985.

[56] Mitsuo Gen and Runwei Cheng. *Genetic algorithms and engineering optimization*, volume 7. John Wiley & Sons, 2000.

[57] Manoj Tiwari and Jenny A Harding. *Evolutionary Computing in Advanced Manufacturing*, volume 73. John Wiley & Sons, 2011.

[58] Runwei Cheng, Mitsuo Gen, and Yasuhiro Tsujimura. A tutorial survey of job-shop scheduling problems using genetic algorithms—i. representation. *Computers & industrial engineering*, 30(4):983–997, 1996.

[59] Victor Oduguwa, Ashutosh Tiwari, and Rajkumar Roy. Evolutionary computing in manufacturing industry: an overview of recent applications. *Applied Soft Computing*, 5(3):281–299, 2005.

[60] Mitsuo Gen, Lin Lin, and Haipeng Zhang. Evolutionary techniques for optimization problems in integrated manufacturing system: State-of-the-art-survey. *Computers & industrial engineering*, 56(3):779–808, 2009.

[61] Vlasis K Koumousis and Christos P Katsaras. A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance. *Evolutionary Computation, IEEE Transactions on*, 10(1):19–28, 2006.

[62] Lin Lin and Mitsuo Gen. Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation. *Soft Computing*, 13(2):157–168, 2009.

[63] Bassem Jarboui, Mansour Eddaly, and Patrick Siarry. An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems. *Computers & Operations Research*, 36(9):2638–2646, 2009.

[64] Quan-Ke Pan and Rubén Ruiz. An estimation of distribution algorithm for lot-streaming flow shop problems with setup times. *Omega*, 40(2):166–180, 2012.

[65] Sheng-yao Wang, Ling Wang, Min Liu, and Ye Xu. An effective estimation of distribution algorithm for solving the distributed permutation flow-shop scheduling problem. *International Journal of Production Economics*, 145(1):387–396, 2013.

[66] Josu Ceberio, Alexander Mendiburu, and Jose A Lozano. Introducing the mallows model on estimation of distribution algorithms. In *Neural Information Processing*, pages 461–470. Springer, 2011.

[67] SH WangFlow-shopFlow shophoi and H Lu. A hybrid estimation of distribution algorithm for simulation-based scheduling in a stochastic permutation flowshop. *Computers & Industrial Engineering*, 90:186–196, 2015.

[68] Sheng-yao Wang, Ling Wang, Min Liu, and Ye Xu. An order-based estimation of distribution algorithm for stochastic hybrid flow-shop scheduling problem. *International Journal of Computer Integrated Manufacturing*, 28(3):307–320, 2015.

[69] Rui Zhang. A rule-based estimation of distribution algorithm for solving job shop. *JCIT: Journal of Convergence Information Technology*, 6(8):220–227, 2011.

[70] Xiao-juan He, Jian-chao Zeng, Li-fang Wang, and Song-dong Xue. A new estimation of distribution algorithm for solving job shop scheduling problems. *International Information Institute (Tokyo). Information*, 16(7):4391, 2013.

[71] Fuqing Zhao, Zhongshi Shao, Junbiao Wang, and Chuck Zhang. A hybrid differential evolution and estimation of distribution algorithm based on neighbourhood search for job shop scheduling problems. *International Journal of Production Research*, pages 1–22, 2015.

[72] Ling Wang, Shengyao Wang, Ye Xu, Gang Zhou, and Min Liu. A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem. *Computers & Industrial Engineering*, 62(4):917–926, 2012.

[73] Shengyao Wang, Ling Wang, Ye Xu, and Min Liu. An effective estimation of distribution algorithm for the flexible job-shop scheduling problem with fuzzy processing time. *International Journal of Production Research*, 51(12):3778–3793, 2013.

[74] Ricardo Pérez-Rodríguez, S Jöns, Arturo Hernández-Aguirre, and Carlos Alberto-Ochoa. Simulation optimization for a flexible jobshop scheduling problem using an estimation of distribution algorithm. *The International Journal of Advanced Manufacturing Technology*, 73(1-4):3–21, 2014.

[75] Uwe Aickelin, Edmund K Burke, and Jingpeng Li. An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering. *Journal of the Operational Research Society*, 58(12):1574–1585, 2007.

[76] Ling Wang and Chen Fang. An effective estimation of distribution algorithm for the multi-mode resource-constrained project scheduling problem. *Computers & Operations Research*, 39(2):449–460, 2012.

[77] Chen Fang, Rainer Kolisch, Ling Wang, and Chundi Mu. An estimation of distribution algorithm and new computational results for the stochastic resource-constrained project scheduling problem. *Flexible Services and Manufacturing Journal*, 27(4):585–605, 2015.

[78] Martin Pelikan. *Hierarchical Bayesian optimization algorithm*. Springer, 2005.

[79] David Jonathan Coffin and Robert Elliott Smith. Why is parity hard for estimation of distribution algorithms? In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 624–624. ACM, 2007.

[80] Robert B Heckendorn and Alden H Wright. Efficient linkage discovery by limited probing. *Evolutionary computation*, 12(4):517–545, 2004.

[81] Masaharu Munetomo and David H Goldberg. Linkage identification by non-monotonicity detection for overlapping functions. *Evolutionary computation*, 7(4):377–398, 1999.

[82] X. Zhu and W. E Wilhelm. Scheduling and lot sizing with sequence-dependent setup: A literature review. *IIE transactions*, 38(11):987–1007, 2006.

[83] Michael L Pinedo. *Scheduling: theory, algorithms, and systems*. Springer Science & Business Media, 2012.

[84] Jacek Błażewicz, Klaus H Ecker, Günter Schmidt, and Jan Weglarz. *Scheduling in computer and manufacturing systems*. Springer Science & Business Media, 2012.

[85] Simon French. *Sequencing and scheduling: an introduction to the mathematics of the job-shop*, volume 683. Ellis Horwood Chichester, 1982.

[86] Peter Brucker and P Brucker. *Scheduling algorithms*, volume 3. Springer, 2007.

[87] Anant Singh Jain and Sheik Meeran. A state-of-the-art review of job-shop scheduling techniques. Technical report, Technical report, Department of Applied Physics, Electronic and Mechanical Engineering, University of Dundee, Dundee, Scotland, 1998.

[88] Najib M Najid, Stephane Dauzere-Pérès, and Ali Zaidat. A modified simulated annealing method for flexible job shop scheduling problem. In *Systems, Man and Cybernetics, 2002 IEEE International Conference on*, volume 5, pages 6–pp. IEEE, 2002.

[89] Imed Kacem, Slim Hammadi, and Pierre Borne. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 32(1):1–13, 2002.

[90] Peter Brucker and Rainer Schlie. Job-shop scheduling with multi-purpose machines. *Computing*, 45(4):369–375, 1990.

[91] S. Shakya and J. McCall. Optimization by estimation of distribution with DEUM framework based on markov random fields. *International Journal of Automation and Computing*, 4(3):262–272, 2007.

[92] Luis M. de Campos and Juan F. Huete. A new approach for learning belief networks using independence criteria. *International Journal of Approximate Reasoning*, 24(1):11–37, April 2000.

[93] Xue-wen Chen, G. Anantha, and Xiaotong Lin. Improving bayesian network structure learning with mutual information-based node ordering in the k2 algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):628–640, 2008.

[94] Siddhartha Shakya and John McCall. Optimization by estimation of distribution with deum framework based on markov random fields. *International Journal of Automation and Computing*, 4(3):262–272, 2007.

[95] George Casella and Edward I George. Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

[96] Ling Wang, Shengyao Wang, Ye Xu, Gang Zhou, and Min Liu. A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem. *Comput. Ind. Eng.*, 62(4):917–926, May 2012.

[97] Takeshi Yamada and Ryohei Nakano. Genetic algorithms for job-shop scheduling problems. In *Proceedings of the Modern Heuristics for Decision Support*, page 67–81, 1997.

[98] Pierre Hansen and Nenad Mladenović. Variable neighborhood search. In Fred Glover and Gary A. Kochenberger, editors, *Handbook of Metaheuristics*, number 57 in International Series in Operations Research & Management Science, pages 145–184. Springer US, January 2003.

[99] Paolo Brandimarte. Routing and scheduling in a flexible job shop by tabu search. *Ann Oper Res*, 41(3):157–183, September 1993.

[100] Stéphane Dauzère-Pérès and Jan Paulli. An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search. *Annals of Operations Research*, 70(0):281–306, April 1997.

[101] J Wesley Barnes and John B Chambers. Solving the job shop scheduling problem with tabu search. *IIE transactions*, 27(2):257–263, 1995.

[102] S. Rahnamayan, H.R. Tizhoosh, and M.M.A. Salama. Opposition-based differential evolution. 12(1):64–79.

[103] X.-S. Yang. *Engineering Optimization: An Introduction with Metaheuristic Applications*.

[104] YF Zhang, AN Saravanan, and JYH Fuh. Integration of process planning and scheduling by exploring the flexibility of process planning. *International Journal of Production Research*, 41(3):611–628, 2003.

[105] M Rajkumar, P Asokan, N Anilkumar, and T Page. A grasp algorithm for flexible job-shop scheduling problem with limited resource constraints. *International Journal of Production Research*, 49(8):2409–2423, 2011.

[106] Mitchell A Potter and Kenneth A De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary computation*, 8(1):1–29, 2000.

[107] Jose A Lozano. *Towards a new evolutionary computation: Advances on estimation of distribution algorithms*, volume 192. Springer Science & Business Media, 2006.

[108] David Maxwell Chickering. Learning bayesian networks is np-complete. In *Learning from data*, pages 121–130. Springer, 1996.

[109] D.M. Chickering. Learning bayesian networks is NP-complete. *Learning from data: Artificial intelligence and statistics v*, 112:121–130, 1996.

[110] JF Lemmer and LN Kanal. Propagating uncertainty in bayesian networks by probabilistic logic sampling. *Uncertainty in Artificial Intelligence 2*, 5:149, 2014.

[111] Serena H Chen and Carmel A Pollino. Guidelines for good practice in bayesian network modelling. In *International Congress on Environmental Modelling and Software Modelling for Environment's Sake, Fifth Biennial Meeting. International Environmental Modelling and Software Society (iEMSs), Ottawa, Canada*, pages 170–178, 2010.

[112] F. Zhang and A.Y.C. Nee. Using genetic algorithms in process planning for job shop machining. *Evolutionary Computation, IEEE Transactions on*, 1(4):278–289, 1997.

[113] W. D. Li, S. K. Ong, and A. Y. C. Nee. Hybrid genetic algorithm and simulated annealing approach for the optimization of process plans for prismatic parts. *International Journal of Production Research*, 40(8):1899, 2002.

[114] W. D. Li and C. A. McMahon. A simulated annealing-based optimization approach for integrated process planning and scheduling. *International Journal of Computer Integrated Manufacturing*, 20(1):80, 2007.

[115] Mojtaba Salehi and Ardeshir Bahreininejad. Optimization process planning using hybrid genetic algorithm and intelligent search for job shop machining. *Journal of intelligent manufacturing*, 22(4):643–652, 2011.

[116] Y.W. Guo, W.D. Li, A.R. Mileham, and G.W. Owen. Applications of particle swarm optimisation in integrated process planning and scheduling. *Robotics and Computer-Integrated Manufacturing*, 25(2):280–288, April 2009.

Index